

**Transcodage entre indices de dictionnaires multi-impulsionnels utilisés
en codage en compression de signaux numériques**

La présente invention concerne le codage/décodage de signaux numériques,
notamment dans des applications de transmission ou de stockage des signaux
multimédias tels que les signaux audio (parole et/ou sons).

Dans le domaine du codage en compression, de nombreux codeurs
modélisent un signal de L échantillons par des impulsions en nombre restreint,
très inférieur au nombre total d'échantillons. C'est, par exemple, le cas de
certains codeurs audio fréquentiels tels que le codeur audio dit "TDAC" décrit
notamment dans le document publié US-2001/027393. Dans ce codeur, les
coefficients de transformée en cosinus discrète modifiée normalisés dans
chaque bande sont quantifiés par des quantificateurs vectoriels utilisant des
dictionnaires algébriques imbriqués en taille, ces codes algébriques
comportent en général quelques composantes non nulles, les autres
composantes étant égales à zéro. C'est aussi le cas de la majorité des
codeurs de parole à analyse par synthèse notamment de type "ACELP" (pour
"Algebraic Code Excited Linear Prediction"), de type MP-MLQ (pour
"MultiPulse Maximum Likelihood Quantization"), ou autres. En effet, pour
modéliser le signal d'innovation, ces codeurs utilisent un répertoire, composé
de formes d'ondes ayant très peu de composantes non nulles dont les
positions et les amplitudes obéissent, de plus, à des règles prédéterminées.

On décrit brièvement ci-après de tels codeurs à analyse par synthèse.

Dans ces codeurs à analyse par synthèse, le modèle de synthèse est utilisé
au codage pour extraire les paramètres modélisant les signaux à coder. Ces
signaux peuvent être échantillonnés à la fréquence téléphonique ($F_e = 8$ kHz)
ou à une fréquence plus élevée, par exemple à 16 kHz pour le codage en
bande élargie (bande passante de 50 Hz à 7 kHz). Selon l'application et la

qualité désirée, le taux de compression varie de 1 à 16. Ces codeurs fonctionnent à des débits de 2 à 16 kbit/s en bande téléphonique, et à des débits de 6 à 32 kbit/s en bande élargie.

5 Le principe d'un dispositif de codage/décodage numérique de type CELP, qui est le codeur/décodateur à analyse par synthèse le plus utilisé actuellement pour le codage/décodage des signaux de parole, est brièvement décrit ci-après. Le signal de parole est échantillonné et converti en une suite de blocs de L' échantillons appelés trames. En général, chaque trame est découpée en
10 blocs plus petits de L échantillons, appelés sous-trames. Chaque bloc est synthétisé en filtrant une forme d'onde extraite d'un répertoire (appelé aussi dictionnaire), multipliée par un gain, à travers deux filtres variant dans le temps. Le dictionnaire d'excitation est un ensemble fini de formes d'ondes de L échantillons. Le premier filtre est le filtre de prédiction à long terme. Une
15 analyse dite "*LTP*" (pour "*Long Term Prediction*") permet d'évaluer les paramètres de ce filtre de prédiction à long terme qui exploite la périodicité des sons voisés (représentant typiquement la fréquence du fondamental (fréquence de vibration des cordes vocales)). Le second filtre est le filtre de prédiction à court terme. Les méthodes d'analyse "*LPC*" (pour "*Linear Prediction Coding*") permettent d'obtenir ces paramètres de prédiction à court
20 terme, représentatifs de la fonction de transfert du conduit vocal et caractéristiques du spectre du signal (représentant typiquement les modulations dues à la forme prise par les lèvres, à la position de la langue et du larynx, ou autres).

25 Le procédé utilisé pour déterminer la séquence d'innovation est la méthode d'analyse par synthèse. Au niveau du codeur, un grand nombre de séquences d'innovation du dictionnaire d'excitation sont filtrées par les deux filtres LTP et LPC, et la forme d'onde sélectionnée est celle produisant le signal synthétique
30 le plus proche du signal original selon un critère de pondération perceptuelle, connu en général sous le nom de critère CELP.

On décrit brièvement ci-après l'utilisation de dictionnaires multi-impulsionnels dans de tels codeurs à analyse par synthèse, étant bien entendu toutefois que les codeurs CELP, comme les décodeurs CELP, sont bien connus de l'homme de l'art.

Le codeur multi-débits de la norme UIT-T G.723.1 est un bon exemple de codeur à analyse par synthèse utilisant des dictionnaires multi-impulsionnels. Ici, les positions d'impulsions sont toutes distinctes. Les deux débits du codeur (6,3 kbit/s et 5,3 kbit/s) modélisent le signal d'innovation par des formes d'onde extraites de dictionnaires qui ne comportent qu'un nombre restreint d'impulsions non nulles: 6 ou 5 pour le haut débit, 4 pour le bas débit. Ces impulsions sont d'amplitude +1 ou -1. Dans son mode à 6,3 kbit/s, le codeur G.723.1 utilise alternativement deux dictionnaires :

- dans le premier utilisé pour les sous-frames paires, les formes d'ondes comportent 6 impulsions et,
- dans le deuxième utilisé pour les sous-frames impaires, elles comportent 5 impulsions.

Pour ces deux dictionnaires, une seule restriction est imposée aux positions des impulsions de tout vecteur-code. Ces positions doivent toutes avoir la même parité, c'est-à-dire qu'elles sont toutes paires ou toutes impaires. Dans le dictionnaire du mode à 5,3 kbit/s, les positions des 4 impulsions sont plus contraintes. Outre la même contrainte de parité que les dictionnaires du mode haut débit, chaque impulsion a un choix limité de positions.

Le dictionnaire multi-impulsionnel du mode à 5,3 kbit/s appartient à la famille bien connue des dictionnaires ACELP. La structure d'un répertoire ACELP est basée sur la technique ISPP (pour "*Interleaved Single-Pulse Permutation*") qui consiste à diviser l'ensemble des L positions en K pistes entrelacées, chacune des N impulsions étant localisée dans certaines pistes prédéfinies. Dans certaines applications, la dimension L des mots de codes peut être étendue à

L+N. Ainsi dans le cas du répertoire du mode bas débit du codeur selon la norme UIT-T G.723.1, la dimension du bloc de 60 échantillons a été étendue à 64 échantillons et les 32 positions paires (respectivement impaires) ont été divisées en 4 pistes entrelacées de longueur 8 ne se recouvrant pas. Il y a donc deux groupes de 4 pistes, un pour chaque parité. La table 1 ci-après présente pour chaque impulsion notée i_0 à i_3 l'ensemble de ces 4 pistes pour les positions paires.

Impulsion	Signe	Positions
i_0	± 1	0, 8, 16, 24, 32, 40, 48, 56
i_1	± 1	2, 10, 18, 26, 34, 42, 50, 58
i_2	± 1	4, 12, 20, 28, 36, 44, 52, (60)
i_3	± 1	6, 14, 22, 30, 38, 46, 54, (62)

Table 1: Positions et amplitudes des impulsions du dictionnaire ACELP du codeur G.723.1 à 5,3 kbit/s

10

Les dictionnaires d'innovation ACELP sont utilisés dans de nombreux codeurs à analyse par synthèse qui sont normalisés (UIT-T G.723.1, UIT-T G.729, IS-641, 3GPP NB-AMR, 3GPP WB-AMR). Les tables 2 à 4 ci-après présentent quelques exemples de ces dictionnaires ACELP pour une longueur de bloc de 40 échantillons. On notera que la contrainte de parité n'est pas utilisée dans ces dictionnaires. La table 2 présente le dictionnaire ACELP à 17 bits et 4 impulsions non nulles d'amplitude ± 1 , utilisé dans le codeur UIT-T G.729 à 8 kbit/s, dans le codeur IS-641 à 7,4 kbit/s ainsi que dans les modes 7,4 et 7,95 kbit/s du codeur 3GPP NB-AMR.

15

Impulsion	Signe	Positions
i_0	± 1	0, 5, 10, 15, 20, 25, 30, 35
i_1	± 1	1, 6, 11, 16, 21, 26, 31, 36
i_2	± 1	2, 7, 12, 17, 22, 27, 32, 37
i_3	± 1	3, 8, 13, 18, 23, 28, 33, 38 4, 9, 14, 19, 24, 29, 34, 39

20

Table 2: Positions et amplitudes des impulsions du dictionnaire ACELP des codeurs UIT-T G.729 à 8 kbit/s, IS641 à 7,4 kbit/s et 3GPP NB-AMR à 7,4 et 7,95 kbit/s

Dans le dictionnaire ACELP à 35 bits utilisé dans le mode 12,2 kbit/s du codeur 3GPP NB-AMR, chaque vecteur-code contient 10 impulsions non nulles d'amplitude ± 1 . Le bloc de 40 échantillons est divisé en 5 pistes de longueur 8, chacune contenant 2 impulsions. On note que les deux impulsions d'une même piste peuvent se recouvrir et résulter en une seule impulsion d'amplitude ± 2 . Ce dictionnaire est présenté à la table 3.

Impulsion	Signe	Positions
i_0, i_5	± 1	0, 5, 10, 15, 20, 25, 30, 35
i_1, i_6	± 1	1, 6, 11, 16, 21, 26, 31, 36
i_2, i_7	± 1	2, 7, 12, 17, 22, 27, 32, 37
i_3, i_8	± 1	3, 8, 13, 18, 23, 28, 33, 38
i_4, i_9	± 1	4, 9, 14, 19, 24, 29, 34, 39

Table 3: Positions et amplitudes des impulsions du dictionnaire ACELP du codeur 3GPP NB-AMR à 12,2 kbit/s

Enfin, la table 4 présente le dictionnaire ACELP à 11 bits et 2 impulsions non nulles d'amplitude ± 1 , utilisé dans l'extension à bas débit (6,4 kbit/s) du codeur UIT-T G.729 et dans le mode 5,9 kbit/s du codeur 3GPP NB-AMR.

Impulsion	Signe	Positions
i_0	± 1	1, 3, 6, 8, 11, 13, 16, 18, 21, 23, 26, 28, 31, 33, 36, 38
i_1	± 1	0, 1, 2, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, 16, 17, 19, 20, 21, 22, 24, 25, 26, 27, 29, 30, 31, 32, 34, 35, 36, 37, 39

Table 4: Positions et amplitudes des impulsions du dictionnaire ACELP des codeurs UIT-T G.729 à 6,4 kbit/s et 3GPP NB-AMR à 5,9 kbit/s

On décrit ci-après ce que l'on entend par "l'exploration" des dictionnaires multi-impulsionnels.

Comme pour toute opération de quantification, la recherche de la modélisation optimale d'un vecteur à coder consiste à choisir dans l'ensemble (ou dans un sous-ensemble) des vecteurs-codes du dictionnaire, celui qui lui "*ressemble*" le plus, c'est-à-dire celui qui minimise une mesure de distance entre lui et ce vecteur d'entrée. Pour ce faire, on procède à une étape dite "*d'exploration*" des dictionnaires.

Dans le cas des dictionnaires multi-impulsionnels, cette étape revient à rechercher la combinaison d'impulsions qui optimise la proximité entre le signal à modéliser et le signal résultant du choix des impulsions. Selon la taille et/ou la structure du dictionnaire, cette exploration peut être exhaustive ou non (donc plus ou moins complexe).

Les dictionnaires utilisés dans le codeur TDAC précité étant des unions de codes à permutation de type II, l'algorithme de codage d'un vecteur de coefficients de transformée normalisés exploite cette propriété pour déterminer son plus proche voisin parmi tous les vecteurs-codes tout en ne calculant qu'un nombre limité de critères de distance (avec une utilisation de vecteurs dits "*leaders absolus*").

Dans les codeurs à analyse par synthèse, l'exploration des dictionnaires multi-impulsionnels n'est pas exhaustive sauf pour les dictionnaires de faible taille. Pour les dictionnaires à plus haut débit, seul un petit pourcentage du dictionnaire est exploré. Par exemple, l'exploration des dictionnaires multi-impulsionnels de type ACELP est généralement effectuée en deux étapes. Pour simplifier cette recherche, une première étape présélectionne pour chaque position possible d'impulsion son amplitude (donc son signe comme indiqué ci-avant) par une simple quantification d'un signal dépendant du signal d'entrée. Les amplitudes des impulsions étant fixées, les positions des impulsions sont alors recherchées par une technique d'analyse par synthèse (selon le critère CELP). En dépit de l'exploitation de la structure en ISPP et du

nombre restreint d'impulsions, une recherche exhaustive des combinaisons des positions n'est effectuée que pour les dictionnaires bas débit (typiquement inférieur ou égal à 12 bits). C'est le cas par exemple, pour le dictionnaire ACELP à 11 bits (table 4) utilisé dans le codeur G.729 à 6,4 kbit/s où les 512 combinaisons de positions de 2 impulsions sont toutes testées pour sélectionner la meilleure, ce qui revient à calculer les 512 critères CELP correspondants.

Pour les dictionnaires à plus haut débit, diverses méthodes dites de focalisation ont été proposées. On parle alors de "*recherche focalisée*".

Certaines de ces méthodes, connues, sont utilisées dans les codeurs normalisés cités ci-avant. Leur but est de réduire le nombre de combinaisons de positions à explorer en se basant sur les propriétés du signal à modéliser. Par exemple, on peut citer l'algorithme dit "*depth-first tree*", utilisé par beaucoup de codeurs ACELP standardisés. Dans cet algorithme, on privilégie certaines positions telles que les maxima locaux des pistes d'un signal-cible dépendant du signal d'entrée, du signal synthétique passé et du filtre composé des filtres de synthèse et de pondération perceptuelle. Il en existe plusieurs variantes selon la taille du dictionnaire utilisé. Pour explorer le dictionnaire ACELP à 35 bits et 10 impulsions (table 3), la première impulsion est placée à la même position que le maximum global du signal-cible. Puis, quatre itérations sont effectuées par permutation circulaire des pistes consécutives. A chaque itération, la position de la deuxième impulsion est fixée au maximum local d'une des 4 autres pistes, les positions des huit autres impulsions restantes sont recherchées séquentiellement par paires dans des boucles imbriquées. On teste à chaque itération 256 (=4 paires×8×8) combinaisons différentes, ce qui permet donc de n'explorer que 1024 combinaisons de positions des 10 impulsions parmi les 2^{25} du dictionnaire. Une autre variante est utilisée dans le codeur IS641 où un pourcentage plus élevé de combinaisons du dictionnaire à 17 bits et 4 impulsions (table 2) est exploré. On

5 teste 768 combinaisons parmi les 8192 ($=2^{13}$) combinaisons de positions d'impulsions. Dans le codeur G.729 à 8 kbit/s, ce même dictionnaire ACELP est exploré par une méthode différente de focalisation. L'algorithme effectue une recherche itérative en imbriquant quatre boucles de recherche des impulsions (une par impulsion). La recherche est focalisée en rendant l'entrée dans la boucle intérieure (recherche de la dernière impulsion appartenant aux pistes 3 ou 4) conditionnelle au dépassement d'un seuil adaptatif. Ce seuil dépend aussi des propriétés du signal-cible (maxima locaux et moyennes des 3 premières pistes). De plus, le nombre maximal d'explorations de combinaisons de 4 impulsions est fixé à 1440 (soit 17,6 % des 8192 combinaisons).

15 Dans le codeur G.723.1 à 6,3 kbit/s, toutes les $2 \times 2^5 \times C_{30}^5$ (respectivement $2 \times 2^6 \times C_{30}^6$) combinaisons de 5 (respectivement 6) impulsions ne sont pas explorées. Pour chaque grille, l'algorithme emploie une analyse connue, de type "*multipulse*", pour rechercher séquentiellement les positions et les amplitudes des impulsions. Comme pour les dictionnaires ACELP, des variantes existent pour restreindre le nombre de combinaisons testées.

20 De telles techniques présentent toutefois les problèmes ci-après.

25 L'exploration, même sous-optimale, d'un dictionnaire multi-impulsionnel constitue dans beaucoup de codeurs une opération coûteuse en temps de calcul. Par exemple, dans les codeurs G.723.1 à 6,3kbit/s et le G.729 à 8 kbit/s, cette recherche représente près de la moitié de la complexité totale du codeur. Pour le NB-AMR, elle représente le tiers de la complexité totale du codeur. Pour le codeur TDAC, elle représente le quart de la complexité totale du codeur.

On comprendra en particulier que cette complexité devient critique lorsque plusieurs codages doivent être effectués par une même unité de traitement, telle qu'une passerelle chargée de gérer de nombreuses communications en parallèle ou un serveur distribuant de nombreux contenus multimédias. Le problème de la complexité est encore accru par la multiplicité des formats de compression qui circulent sur les réseaux.

En effet, pour offrir mobilité et continuité, les services de communication multimédias modernes et innovants doivent pouvoir fonctionner dans une grande variété de conditions. Le dynamisme du secteur de la communication multimédia et l'hétérogénéité des réseaux, accès et terminaux ont engendré une prolifération de formats de compression dont la présence dans les chaînes de communication nécessite plusieurs codages soit en cascade (transcodage), soit en parallèle (codage multi-format ou codage multi-mode).

On définit ci-après ce que l'on entend par "*transcodage*". Le transcodage devient nécessaire lorsque, dans une chaîne de transmission, une trame de signal compressée émise par un codeur ne peut plus poursuivre son chemin, sous ce format. Le transcodage permet de convertir cette trame sous un autre format compatible avec la suite de la chaîne de transmission. La solution la plus élémentaire (et la plus courante à l'heure actuelle) est la mise bout à bout d'un décodeur et d'un codeur. La trame compressée arrive sous un premier format, elle est décompressée. Ce signal décompressé est alors recompressé sous un second format accepté par la suite de la chaîne de communication. Cette mise en cascade d'un décodeur et d'un codeur est appelée "*tandem*". Cette solution est très coûteuse en complexité (essentiellement à cause du recodage) et elle dégrade la qualité. En effet, le second codage se fait sur un signal décodé, qui est une version dégradée du signal original. Par ailleurs, une trame peut rencontrer plusieurs tandems avant d'arriver à destination. On imagine aisément le coût en calcul et la perte de qualité. De plus, les retards

liés à chaque opération de tandem s'accumulent et peuvent nuire à l'interactivité des communications.

5 Par ailleurs, la complexité pose aussi problème dans le cadre d'un système de compression multi-format où un même contenu est compressé sous plusieurs formats. C'est le cas des serveurs de contenus qui diffusent un même contenu sous plusieurs formats adaptés aux conditions d'accès, de réseaux et terminaux des différents clients. Cette opération de multi-codage devient extrêmement complexe à mesure qu'augmente le nombre de formats désirés,
10 ce qui sature rapidement les ressources du système.

Un autre cas de codage multiple en parallèle est la compression multi-mode à décision a posteriori. A chaque segment de signal à coder, plusieurs modes de compression sont effectués et celui qui optimise un critère donné ou obtient le
15 meilleur compromis débit/distorsion est sélectionné. Là encore, la complexité de chacun des modes de compression en limite le nombre et/ou conduit à élaborer une sélection a priori d'un nombre très restreint de modes.

On décrit ci-après les approches, connues, qui ont été mises en œuvre pour
20 tenter de résoudre les problèmes posés décrits ci-avant.

Les nouvelles applications de communication multimédia (tels que l'audio et la vidéo) nécessitent souvent plusieurs codages soit en cascade (transcodage) soit en parallèle (multi-codage et codage multi-mode à décision a posteriori).
25 La barrière de complexité posée par tous ces codages reste un problème à résoudre, malgré l'augmentation des puissances de traitement actuelles. La plupart de ces opérations connues de codage multiple ne prennent pas en compte les interactions entre les formats et entre le format de E et son contenu. Cependant, quelques techniques de transcodage intelligent ont été
30 proposées qui ne se contentent plus de décoder puis recoder, mais exploitent

les similitudes entre formats de codages et permettent ainsi de réduire la complexité tout en limitant la dégradation apportée.

On décrit ci-après des méthodes de transcodage dit "*intelligent*".

- 5 Au sein d'une même famille de codeurs (CELP, paramétrique, par transformée, ou autre), tous les codeurs extraient du signal les mêmes paramètres physiques. Pourtant, il existe une grande variété de modélisation et/ou quantification de ces paramètres. Ainsi, d'un codeur à un autre, un même paramètre peut être codé de la même manière, ou, au contraire, très
10 différemment.

- Par ailleurs, les codages peuvent être strictement identiques. Ils peuvent être identiques dans la modélisation et le calcul du paramètre, mais se différencier simplement par la traduction du codage sous forme de bits. Enfin, ils peuvent
15 être complètement différents tant par la modélisation du paramètre que sa quantification, voire par sa fréquence d'analyse ou d'échantillonnage.

- Si la modélisation et le calcul d'un paramètre sont strictement identiques, y compris dans la traduction binaire, il suffit de recopier le champ binaire
20 correspondant du train binaire généré par le premier codeur vers celui du second. Ce cas très favorable se présente par exemple lors d'un transcodage de la norme G.729 à la norme IS-641 pour l'excitation adaptative (retards LTP).

- 25 Si, pour le même paramètre, les deux codeurs ne se distinguent que par la traduction binaire du paramètre calculé, il suffit de décoder le champ binaire du premier format, puis de repasser dans le domaine binaire en utilisant la méthode de codage selon le second format. Cette conversion peut aussi être effectuée par des tables de correspondance bijective. C'est le cas par exemple

lors d'un transcodage des excitations fixes de la norme G.729 à la norme AMR (7,4 et 7,95kbit/s).

5 Dans les deux cas précédents, le transcodage du paramètre se fait en restant au niveau des bits. Une simple manipulation de bits permet de rendre le paramètre compatible avec le second format de codage. En revanche, lorsqu'un paramètre extrait du signal est modélisé ou quantifié différemment par deux formats de codage, le passage de l'un vers l'autre ne se fait pas aussi simplement. Plusieurs méthodes ont été proposées. Elles opèrent soit
10 au niveau du paramètre, soit au niveau de l'excitation, soit au niveau du signal décodé.

Pour le transcodage dans le domaine des paramètres, rester au niveau du paramètre est possible lorsque les deux formats de codage calculent un
15 paramètre de la même manière mais le quantifient différemment. Les différences de quantifications peuvent être liées à la précision choisie ou encore à la méthode choisie (scalaire, vectorielle, prédictive, ou autre). Il suffit alors de décoder le paramètre, puis de le quantifier par la méthode du second format de codage. Cette méthode connue est appliquée actuellement,
20 notamment pour le transcodage des gains d'excitations. Souvent, il faut modifier le paramètre décodé avant sa requantification. Par exemple, si les codeurs ont des fréquences d'analyse du paramètre ou des longueurs de trames/sous-trames différentes, il est courant d'interpoler/décimer les paramètres. L'interpolation peut se faire par exemple selon la méthode décrite
25 dans le document publié US2003/033142. Une autre modification possible est d'arrondir le paramètre à la précision qui lui est imposée par le second format de codage. Ce cas se présente surtout pour la hauteur de la fréquence fondamentale (ou "*pitch*").

30 S'il n'est pas possible de transcoder un paramètre en restant dans le domaine du paramètre, on peut remonter à un niveau supérieur dans le décodage. Il

s'agit du domaine de l'excitation, sans pour autant aller jusqu'au domaine du signal. Cette technique a été proposée pour les gains dans la référence :

5 *"Improving transcoding capability of speech coders in clean and frame erasured channel environments"* de Hong-Goo Kang, Hong Kook Kim, Cox, R.V., dans Speech Coding, 2000, Proceedings 2000, IEEE Workshop on Speech Coding, Pages 78–80.

10 Enfin, une dernière solution (la plus complexe et la moins *"intelligente"*) consiste à recalculer explicitement le paramètre comme le ferait le codeur, mais à partir d'un signal synthétisé. Cette opération revient à une sorte de tandem partiel, seuls certains paramètres étant entièrement recalculés. Cette méthode a été appliquée à divers paramètres tels que l'excitation fixe, les gains dans la référence IEEE précitée, ou le pitch.

15 Pour le transcodage des impulsions, même si plusieurs techniques ont été développées afin de calculer les paramètres rapidement et à moindre coût, peu de ces solutions aujourd'hui ont une approche intelligente pour calculer les impulsions d'un format à partir du paramètre équivalent sous un autre format. En codage à analyse par synthèse, le transcodage intelligent des codes
20 impulsionnels n'est appliqué que lorsque la modélisation est identique (ou proche). En revanche, si les modélisations sont différentes, la méthode du tandem partiel est utilisée. Il faut noter que pour limiter la complexité de cette opération, des approches focalisées exploitant les propriétés du signal décodé ou d'un signal dérivé tel qu'un signal-cible ont été proposées. Dans le
25 document précité US-2001/027393, dans l'exemple de réalisation mettant en œuvre un codeur par transformée MDCT, est présentée une procédure de changement de débit qui peut être considérée comme un cas particulier du transcodage intelligent. En effet, cette procédure permet de requantifier un vecteur d'un premier dictionnaire par un vecteur d'un second dictionnaire. Elle
30 distingue pour cela deux cas selon l'appartenance ou non du vecteur à requantifier dans le deuxième dictionnaire. Si le vecteur quantifié appartient au

nouveau dictionnaire, la modélisation est identique ; sinon, la méthode du décodage partiel est appliquée.

5 La présente invention propose, en se démarquant de toutes ces techniques connues, un transcodage multi-impulsionnel basé sur une sélection d'un sous-ensemble de combinaisons de positions d'impulsions d'un ensemble de jeux d'impulsions à partir d'une combinaison de positions d'impulsions d'un autre ensemble de jeux d'impulsions, les deux ensembles pouvant se distinguer par le nombre d'impulsions qu'ils comportent ainsi que par les règles régissant
10 leurs positions et/ou leurs amplitudes. Ce transcodage est très utile notamment pour les codages multiples en cascade (transcodages) ou en parallèle (multi-codages et codages multi-modes).

15 A cet effet, la présente invention propose tout d'abord un procédé de transcodage entre un premier codeur/décodeur en compression et au moins un second codeur/décodeur en compression. Ces premier et second codeurs/décodeurs sont de type impulsionnel et utilisent des dictionnaires multi-impulsionnels dans lesquels chaque impulsion comporte une position repérée par un indice associé.

20 Le procédé de transcodage au sens de l'invention comporte les étapes suivantes :

- a) le cas échéant, adaptation de paramètres de codage entre lesdits premier et second codeurs/décodeurs,
- 25 b) obtention, à partir du premier codeur/décodeur, d'un nombre choisi de positions d'impulsions et d'indices de positions respectivement associés,
- c) pour chaque position d'impulsion courante d'indice donné, formation d'un groupe de positions d'impulsions comportant au moins la position d'impulsion courante et des positions d'impulsions d'indices associés immédiatement
30 inférieurs et immédiatement supérieurs à l'indice donné,

d) sélection, en fonction de positions d'impulsions admises par le second codeur/décodeur, d'une partie au moins des positions d'impulsions dans un ensemble constitué par une union desdits groupes formés à l'étape c), et
e) transmission des positions des impulsions ainsi sélectionnées au second codeur/décodeur, pour un codage/décodage à partir desdites positions transmises.

Ainsi, la sélection de l'étape d) fait intervenir un nombre de positions d'impulsions possibles inférieur au nombre total de positions d'impulsions possibles du dictionnaire du second codeur/décodeur.

On comprendra en particulier qu'à l'étape e), dans le cas où le second codeur/décodeur précité est un codeur, les positions d'impulsion sélectionnées sont transmises à ce codeur, pour un codage par recherche uniquement parmi les positions transmises. Dans le cas où le second codeur/décodeur précité est un décodeur, ces positions d'impulsion sélectionnées sont transmises pour un décodage de ces positions.

Préférentiellement, l'étape b) utilise un décodage partiel du flux binaire fourni par le premier codeur/décodeur en vue d'identifier un premier nombre de positions d'impulsions qu'utilise le premier codeur/décodeur, dans un premier format de codage. Ainsi, le nombre choisi à l'étape b) correspond préférentiellement à ce premier nombre de positions d'impulsions.

Dans une réalisation avantageuse, les étapes ci-avant sont mises en œuvre par un produit programme d'ordinateur comportant des instructions de programme à cet effet. A ce titre la présente invention vise aussi un tel produit programme d'ordinateur, destiné à être stocké dans une mémoire d'une unité de traitement, notamment d'un ordinateur ou d'un terminal mobile, ou sur un support mémoire amovible et destiné à coopérer avec un lecteur de l'unité de traitement.

La présente invention vise aussi un dispositif de transcodage entre des premier et second codeurs/décodeurs en compression, et comportant alors une mémoire propre à stocker des instructions d'un produit programme d'ordinateur du type décrit ci-avant.

D'autres caractéristiques et avantages de l'invention apparaîtront à l'examen de la description détaillée ci-après, et des dessins annexés sur lesquels :

- la figure 1a représente schématiquement le contexte de transcodage au sens de la présente invention, dans une configuration "*en cascade*",
- la figure 1b représente schématiquement le contexte de transcodage au sens de la présente invention, dans une configuration "*en parallèle*",
- la figure 2 représente schématiquement les différents cas prévus pour les traitements de transcodage à effectuer,
- la figure 2a représente schématiquement un traitement d'adaptation prévu pour le cas où les fréquences d'échantillonnage des premier E et second S codeurs sont différentes,
- la figure 2b représente schématiquement une variante du traitement de la figure 2a,
- la figure 3 résume les étapes du procédé de transcodage au sens de l'invention,
- la figure 4 représente schématiquement le cas de deux sous-frames respectivement des codeurs E et S, de durées différentes L_e et L_s (avec $L_e > L_s$), mais de mêmes fréquences d'échantillonnage,
- la figure 4b représente à titre d'exemple un cas pratique de la figure 4 en illustrant la correspondance temporelle entre un codeur G.723.1 et un codeur G.729,
- la figure 5 illustre schématiquement le découpage de l'excitation du premier codeur E au rythme du second codeur S,
- la figure 6 illustre le cas où l'une des pseudo-sous-frames STE'0 est vide,

la figure 7 représente schématiquement un traitement d'adaptation prévu pour le cas où les durées de sous-trame des premier E et second S codeurs sont différentes.

5 On indique tout d'abord que la présente invention s'inscrit dans la modélisation et le codage des signaux numériques multimédia tels que les signaux audio (parole et/ou sons) par des dictionnaires multi-impulsionnels. Elle peut être mise en œuvre dans le cadre du codage/décodage multiple en cascade ou en parallèle ou de tout autre système utilisant la modélisation d'un signal par une
10 représentation multi-impulsionnelle et qui, à partir de la connaissance d'un premier jeu d'impulsions appartenant à un premier ensemble, doit déterminer au moins un jeu d'impulsions d'un second ensemble. Par souci de concision, seul le cas d'un passage d'un premier ensemble vers un autre ensemble est décrit mais l'invention s'applique aussi dans le cas de passage vers n ($n \geq 2$)
15 ensembles. Par ailleurs, on ne décrit ci-après que le cas d'un "transcodage" entre deux codeurs, mais, bien entendu, le transcodage entre un codeur et un décodeur, s'en déduit sans difficulté majeure.

On considère donc le cas de deux modélisations d'un signal par des jeux
20 d'impulsions correspondant à deux systèmes de codage. Sur les figures 1a et 1b, on a représenté un dispositif de transcodage D entre un premier codeur E, utilisant un premier format de codage COD1, et un second codeur S, utilisant un second format de codage COD2. Le codeur E délivre un flux (ou "train") binaire codé s_{CE} (sous la forme d'une succession de trames codées) au
25 dispositif de transcodage D, lequel comporte un module 10 de décodage partiel pour récupérer le nombre N_e de positions d'impulsions utilisées dans le premier format de codage et les positions p_e de ces impulsions. Comme on le verra en détail ci-après, le dispositif de transcodage au sens de l'invention procède à une extraction des voisinages de droite v_d^e et de gauche v_g^e de
30 chaque position d'impulsion p_e et sélectionne, dans l'union de ces voisinages, des positions d'impulsions qui seront reconnues par le second codeur S. Le

module 11 du dispositif de transcodage représenté sur les figures 1a et 1b effectue donc ces étapes pour délivrer cette sélection de positions (notée S_j sur ces figures 1a et 1b) au second codeur S. On comprendra en particulier qu'à partir de cette sélection S_j , on constitue un sous-répertoire de taille inférieure à la taille du dictionnaire qu'utilise habituellement le second codeur S, selon un avantage que procure l'invention. En utilisant ce sous-répertoire, le codage qu'effectue le codeur S est bien entendu plus rapide car plus restreint, sans nuire pour autant à la qualité de codage.

10 Dans l'exemple représenté sur la figure 1a, le dispositif de transcodage D comporte en outre un module 12 de décodage au moins partiel du flux codé s_{CE} que délivre le premier codeur E. Le module 12 fournit alors au second codeur S une version au moins partiellement décodée s'_0 du signal d'origine s_0 . Le second codeur S délivre ensuite, sur la base de cette version s'_0 , un flux binaire codé s_{CS} .

Dans cette configuration, le dispositif de transcodage D effectue donc une adaptation de codage entre le premier codeur E et le second codeur S, en favorisant avantageusement un codage plus rapide (car plus restreint) auprès du second codeur S. Bien entendu, en variante, l'entité référencée S sur les figures 1a et 1b peut être un décodeur et, dans cette variante, le dispositif D au sens de l'invention effectue un transcodage, proprement dit, entre un codeur E et un décodeur S, ce décodage s'effectuant rapidement grâce aux informations fournies par le dispositif D. Le processus étant réversible, on comprendra que, de façon beaucoup plus générale, le dispositif de transcodage D au sens de la présente invention opère entre un premier codeur/décodeur E et un second codeur/décodeur S.

On indique que l'agencement du codeur E, du transcodeur D et du codeur S peut respecter une configuration "en cascade", comme représentée sur la figure 1a. Dans la variante représentée sur la figure 1b, cet agencement peut

respecter une configuration "*en parallèle*". Dans ce cas, les deux codeurs E et S reçoivent le signal original s_0 et les deux codeurs E et S délivrent respectivement les flux codés s_{CE} et s_{CS} . Bien entendu, le second codeur S n'a plus à recevoir ici la version s'_0 de la figure 1a et le module de décodage au moins partiel 12 du dispositif de transcodage D n'est plus nécessaire. On indique en outre que, si le codeur E peut fournir une sortie compatible avec l'entrée du module 11 (tant par rapport à son nombre d'impulsions que par rapport à ses positions d'impulsions), le module 10 peut simplement être omis ou "*court-circuité*".

10

On indique en outre que le dispositif de transcodage D peut simplement être équipé d'une mémoire stockant les instructions pour mettre en œuvre les étapes ci-avant et d'un processeur pour traiter ces instructions.

15

20

L'application de l'invention se situe donc comme suit. Le premier codeur E a effectué son opération de codage sur un signal donné s_0 (par exemple le signal original). On dispose donc des positions des impulsions choisies par le premier codeur E. Ce codeur a déterminé ces positions p_e par une technique qui lui est propre lors du processus de codage. Le second codeur S doit réaliser lui aussi son codage. Dans le cas du transcodage, le second codeur S ne dispose que du train binaire généré par le premier codeur et l'invention est ici applicable au transcodage dit "*intelligent*" tel que défini ci-avant. Dans le cas du codage multiple en parallèle, le second codeur S dispose aussi du signal dont dispose le premier codeur et l'invention s'applique ici au "*multi-codage intelligent*". En effet, un système qui désire coder un même contenu sous plusieurs formats peut exploiter les informations d'un premier format pour simplifier les opérations de codage des autres formats. L'invention peut aussi s'appliquer au cas particulier du codage multiple en parallèle qu'est le codage multi-mode à décision a posteriori.

30

La présente invention permet de déterminer rapidement les positions p_s (ou, notée indistinctement encore ci-après, s_i) des impulsions pour un autre format de codage à partir de positions p_e (ou, notée indistinctement encore ci-après, e_i) des impulsions d'un premier format. Elle permet de réduire
5 considérablement la complexité de calcul de cette opération pour le second codeur en limitant le nombre de positions possibles. A cet effet, elle utilise les positions choisies par le premier codeur pour définir un ensemble restreint de positions dans l'ensemble des positions possibles du second codeur, ensemble restreint dans lequel on recherchera le meilleur jeu de positions
10 pour les impulsions. Il en résulte un gain en complexité important tout en limitant la dégradation du signal par rapport à une recherche exhaustive ou focalisée classique.

On comprendra ainsi que la présente invention limite le nombre de positions
15 possibles en définissant un ensemble restreint de positions à partir des positions du premier format de codage. Elle se distingue des solutions existantes dans la mesure où celles-ci utilisent uniquement les propriétés du signal à modéliser pour limiter le nombre de positions possibles, en privilégiant et/ou en éliminant des positions.

20 Préférentiellement, pour chaque impulsion d'un jeu d'un premier ensemble, on définit deux voisinages (un droit et un gauche) de largeur ajustable plus ou moins contrainte et on en extrait un ensemble de positions possibles, dans lequel sera présélectionnée au moins une combinaison d'impulsions
25 respectant les contraintes du deuxième ensemble.

Avantageusement, le procédé de transcodage permet d'optimiser le compromis complexité/qualité en adaptant le nombre de positions d'impulsions et/ou les tailles respectives (en terme de combinaisons de positions
30 d'impulsions) des voisinages droit et gauche, pour chaque impulsion. Cet ajustement peut être effectué au début du traitement ou à chaque sous-trame

en fonction de la complexité autorisée et/ou du jeu de positions de départ. L'invention permet aussi d'ajuster/limiter le nombre de combinaisons de positions en favorisant avantageusement les voisinages immédiats.

5 Comme indiqué ci-avant, la présente invention vise aussi un produit programme d'ordinateur dont l'algorithme est conçu notamment pour l'extraction de positions voisines qui facilite la composition des combinaisons d'impulsions du deuxième ensemble.

10 Comme indiqué précédemment, l'hétérogénéité des réseaux et des contenus peut mettre en présence des formats de codage très variés. Les codeurs peuvent se distinguer par de nombreuses caractéristiques. En particulier, deux d'entre elles déterminent sensiblement le mode de fonctionnement de l'invention. Il s'agit de la fréquence d'échantillonnage et de la durée d'une
15 sous-trame. Ci-après, les différents cas possibles sont présentés en correspondance de mises en œuvre de l'invention selon ces différents cas.

La figure 2 synthétise les différents cas. Dans un premier temps, on obtient :

- les nombres de positions d'impulsions N_e, N_s ,
- 20 - les fréquences d'échantillonnage F_e, F_s respectives,
- et les durées de sous-trame L_e, L_s

qu'utilisent respectivement les codeurs E et S (étape 21).

Ainsi, on comprendra déjà que les étapes d'adaptation et de récupération des nombres de positions d'impulsions N_e, N_s peuvent avantageusement être
25 interverties ou simplement menées simultanément.

Au test 22, on compare les fréquences d'échantillonnage. Si les fréquences sont égales, on compare, au test 23, les durées de sous-trame. Sinon, on adapte les fréquences d'échantillonnage, à l'étape 32, selon un procédé décrit
30 ci-après. A l'issue du test 23, si les durées de sous-trame sont égales, on compare, au test 24, les nombres de positions d'impulsions N_e et N_s utilisés

respectivement par le premier et le second format de codage. Sinon, on adapte les durées de sous-trame à l'étape 33 selon un procédé qui sera aussi décrit ci-après. On comprendra que les étapes 22, 23, 32 et 33 définissent ensemble l'étape a) d'adaptation des paramètres de codage citée ci-avant. On indique que les étapes 22 et 32 (adaptation des fréquences d'échantillonnage), d'une part, et les étapes 23 et 33 (adaptation des durées de sous-trame), d'autre part, peuvent être inversées.

On décrit d'abord ci-après le cas où les fréquences d'échantillonnage sont égales et les durées de sous-frames sont égales.

Ce cas est le plus favorable. Cependant, il faut distinguer encore le cas où le premier format utilise plus d'impulsions que le second ($N_e \geq N_s$), et le cas contraire ($N_e < N_s$), selon le résultat du test 24.

*** Cas $N_e \geq N_s$ de la figure 2**

Le principe est le suivant. En considérant les deux codeurs E et S, leurs répertoires utilisent respectivement N_e et N_s impulsions à chaque sous-trame.

Le codeur E a calculé les positions de ses N_e impulsions sur la sous-trame s_e . On note ci-après e_i (ou, indistinctement, p_e) ces positions. L'ensemble restreint P_s des positions privilégiées pour les impulsions du répertoire du codeur S est alors constitué de N_e positions e_i et de leurs voisinages.

$$P_s = \bigcup_{i=0}^{N_e-1} \left\{ \bigcup_{k=-v_g^i}^{v_d^i} \{e_i + k\} \right\}$$

où v_d^i et $v_g^i \geq 0$ sont les tailles des voisinages droit et gauche de l'impulsion i .

Les valeurs de v_d^i et v_g^i , choisies à l'étape 27 de la figure 2, sont plus ou moins grandes selon la complexité et la qualité désirées. Ces tailles peuvent être fixées arbitrairement au début du traitement, ou être choisies à chaque sous-trame s_e .

A l'étape 29 de la figure 2, l'ensemble P_s contient alors chaque position e_i ainsi que ses v_d^l voisins de droite et ses v_g^l voisins de gauche.

- 5 Pour chacune des N_s impulsions du répertoire du codeur S, il faut ensuite définir les positions que cette impulsion a le droit de prendre parmi celles proposées par P_s .

10 Pour cela, on introduira des règles présidant à la construction du répertoire de S. On suppose que les N_s impulsions de S appartiennent à des sous-ensembles pré-définis de positions, un nombre donné d'impulsions partageant un même sous-ensemble de positions autorisées. Par exemple, les 10 impulsions du mode 12,2 bit/s du codeur 3GPP NB-AMR sont réparties 2 par 2 dans 5 sous-ensembles différents, comme le montre la table 3 donnée ci-

15 avant. On note N'_s le nombre de sous-ensembles de positions différents ($N'_s \leq N_s$ dans cet exemple puisque $N'_s=5$), et T_j (pour $j=1$ à N'_s) les sous-ensembles de positions définissant le répertoire de S.

20 A partir de l'ensemble P_s , les N'_s sous-ensembles S_j issus de l'intersection de P_s avec l'un des ensembles T_j sont constitués, à l'étape 30 de la figure 2, selon la relation :

$$S_j = P_s \cap T_j$$

25 Les voisinages v_d^l et v_g^l doivent être d'une taille suffisante pour qu'aucune intersection ne soit vide. On doit permettre ainsi un réajustement des tailles de voisinage, si nécessaire, en fonction du jeu d'impulsions de départ. C'est l'objet du test 34 de la figure 2, avec une augmentation de la taille des voisinages (étape 35) et retour vers la définition de l'union P_s des groupes formés à l'étape c) (étape 29 sur la figure 2) si l'une des intersections est vide.

Au contraire, si aucune intersection S_j n'est vide, c'est le sous-répertoire constitué par ces intersections S_j qui est envoyé au codeur S (étape de fin 31).

Avantageusement, l'invention exploite la structure des répertoires. Par exemple, si le répertoire du codeur S est de type ACELP, ce sont les intersections des positions des pistes avec P_s qui sont calculées. Si le répertoire du codeur E est aussi de type ACELP, la procédure d'extraction des voisinages exploite aussi la structure en pistes et les deux étapes d'extraction des voisinages et composition des sous-ensembles restreints de positions, sont judicieusement associées. En particulier, il est intéressant que l'algorithme d'extraction des voisinages prenne en compte la composition des combinaisons d'impulsions selon les contraintes du deuxième ensemble. Comme on le verra plus loin, des algorithmes d'extraction de voisinages sont élaborés pour faciliter la composition des combinaisons d'impulsions du deuxième ensemble. Un exemple d'un tel algorithme est illustré par l'un des modes de réalisation donnés plus loin (ACELP à 2 impulsions vers ACELP à 4 impulsions).

Le nombre de combinaisons de positions possibles est ainsi restreint et la taille du sous-ensemble du répertoire du codeur S est généralement très inférieure à celle du répertoire original, ce qui réduit grandement la complexité de l'avant-dernière étape du transcodage. On précise ici que le nombre de combinaisons de positions d'impulsions définit la taille du sous-ensemble précité. On précise en outre que c'est le nombre de positions d'impulsions que l'on diminue au sens de l'invention, ce qui entraîne une diminution du nombre de combinaisons de positions d'impulsions et permet alors d'obtenir un sous-répertoire de taille restreinte.

L'étape référencée 46 sur la figure 3 consiste alors à lancer la recherche du meilleur jeu de positions pour les N_s impulsions dans ce sous-répertoire de taille restreinte. Le critère de sélection est similaire à celui du processus de

codage. Pour réduire davantage la complexité, on peut accélérer l'exploration de ce sous-répertoire en utilisant des techniques de focalisation connues décrites ci-avant.

- 5 La figure 3 résume les étapes de l'invention pour le cas où le codeur E utilise au moins autant d'impulsions que le codeur S. On indique toutefois que, comme on l'a déjà vu en référence à la figure 2, si le nombre de positions N_s au second format (le format de S) est supérieur au nombre de positions N_e au premier format (le format de E), le traitement prévu ne se distingue que par quelques variantes avantageuses qui seront décrites plus loin.
- 10

En bref, ces étapes de la figures 3 se résument comme suit. Après une étape a) éventuelle d'adaptation des paramètres de codage (si nécessaire et représentée à cet effet par des traits pointillés sur la figure 3 dans le bloc 41) :

- 15
- récupération des positions e_i des impulsions du codeur E, et préférentiellement d'un nombre N_e de positions (étape 42 correspondant à l'étape b) précitée),
 - extraction des voisinages et formation des groupes de voisinages selon la relation :

20

$$P_s = \bigcup_{i=0}^{N_e-1} \left\{ \bigcup_{k=-v_g^i}^{v_d^i} \{e_i + k\} \right\}$$

(étape 43 correspondant à l'étape c) précitée)

- 25
- composition des sous-ensembles restreints de positions $\{S_j = P_s \cap T_j\}$ formant la sélection de l'étape d) précitée et correspondant à l'étape 44 représentée sur la figure 3, et
 - et transmission de cette sélection au codeur S (étape 45 correspondant à l'étape e) précitée).

On indique qu'après cette étape 45, le codeur S choisit ensuite un jeu de positions dans le répertoire restreint obtenu à l'étape 44.

Le procédé se poursuit donc par une étape 46 de recherche dans ce sous-répertoire reçu par le codeur S d'un jeu de positions optimal ($\text{opt}(S_i)$) comportant le second nombre N_s de positions, comme indiqué ci-avant. Cette
5 étape 46 de recherche du jeu de positions optimal est mise en œuvre préférentiellement par une recherche focalisée pour accélérer l'exploration du sous-répertoire. Le traitement se poursuit naturellement par le codage qu'effectue ensuite le second codeur S.

- 10 On décrit maintenant les traitements prévus pour le cas où le nombre d'impulsions N_e utilisées par le premier format de codage est inférieur au nombre d'impulsions N_s utilisées par le second format de codage.

*** cas $N_e < N_s$ de la figure 2**

- 15 Si le format de S utilise plus d'impulsions que le format de E, le traitement est similaire au traitement exposé ci-dessus. Cependant, il peut arriver que des impulsions du format de S n'aient pas de positions dans le répertoire restreint. Dans ce cas, dans une première réalisation, on autorise toutes les positions possibles pour ces impulsions. Dans une seconde réalisation, préférée, on
20 augmente simplement la taille des voisinages à V'_d et V'_g , à l'étape 28 de la figure 2.

*** cas $N_e < N_s < 2N_e$ de la figure 2**

- Un cas particulier doit être souligné ici. Si N_e est proche de N_s , typiquement
25 $N_e < N_s < 2N_e$, alors une manière préférée de déterminer les positions est envisageable, bien que le traitement précédent reste encore tout à fait applicable. On peut encore gagner en complexité en fixant directement les positions des impulsions de S à partir de celles de E. En effet, les N_e premières impulsions de S sont placées sur les positions de celles de E. Les
30 $N_s - N_e$ impulsions restantes sont placées le plus près possible (au voisinage immédiat) des N_e premières impulsions. Ainsi, on teste à l'étape 25 de la

figure 2 si les nombres N_e et N_s sont proches (avec $N_e > N_s$) et, dans l'affirmative, on procède comme décrit ci-avant pour le choix des positions d'impulsions à l'étape 26.

5 Bien entendu, dans les deux cas $N_e < N_s$ et $N_e < N_s < 2N_e$, si, malgré ces précautions, l'une des intersections S_j est vide, on augmente simplement la taille des voisinages V_g^*, V_d^* , à l'étape 35, comme décrit dans le cas où $N_e \geq N_s$.

10 Finalement, dans tous les cas, si aucune intersection S_j n'est vide, on transmet le sous-répertoire formé par les S_j au second codeur S (étape 31).

On décrit maintenant les traitements prévus à l'étape d'adaptation a) lorsque les paramètres de codage des premier et second formats ne sont pas les mêmes, en particulier en terme de fréquences d'échantillonnage et de durées
15 de sous-trame.

On distingue alors les cas ci-après.

20 *** Durées de sous-trame égales mais fréquences d'échantillonnages différentes**

Cette situation correspond au cas "n" pour le test 22 et "o" pour le test 23 de la figure 2. L'adaptation a) porte alors sur l'étape 32 de la figure 2.

25 On ne peut plus appliquer directement ici le traitement précédent car les deux formats n'ont pas le même découpage du temps. En effet, du fait des fréquences d'échantillonnage différentes, les deux trames ne comportent pas le même nombre d'échantillons sur une même durée.

30 Plutôt que de déterminer les positions des impulsions du format du codeur S sans tenir compte de celles du format du codeur E, comme le ferait un tandem, on propose ici deux traitements selon deux modes de réalisation distincts. Ces

traitements sont de faible complexité, en établissant une correspondance entre les positions des deux formats qui permet alors de se ramener ensuite au traitement décrit ci-avant (comme si les fréquences d'échantillonnage étaient égales).

5

Le traitement du premier mode de réalisation utilise une quantification directe de l'échelle temporelle du premier format par celle du deuxième format. Cette opération de quantification qui peut être tabulée ou calculée par une formule, permet ainsi de trouver, pour chaque position d'une sous-trame du premier format, son équivalent dans une sous-trame du second format et réciproquement.

10

Par exemple, la correspondance entre les positions p_e et p_s dans les sous-frames des deux formats peut être définie par la formule suivante :

$$15 \quad p_s = \left\lfloor \frac{F_s}{F_e} * p_e + 0,5 \right\rfloor, \quad 0 \leq p_e < L_e \text{ et } 0 \leq p_s < L_s$$

où F_e et F_s sont les fréquences d'échantillonnages respectives de E et S, et L_e et L_s leurs longueurs de sous-trame,

$\lfloor \rfloor$ dénotant la partie entière.

20

Selon les caractéristiques de l'unité de traitement, cette correspondance pourra utiliser la formule ci-avant ou avantageusement être tabulée pour les L_e valeurs. On peut aussi choisir une solution intermédiaire en ne tabulant que les l_e premières valeurs ($l_e = \frac{L_e}{d}$, d étant le plus grand commun diviseur de L_e et L_s), les positions restantes se déduisant alors aisément.

25

Il faut noter qu'on peut aussi faire correspondre plusieurs positions de la sous-trame de S à une position d'une sous-trame de E. Par exemple, en retenant les positions immédiatement inférieure et immédiatement supérieure à $\frac{F_s}{F_e} * p_e$.

A partir de l'ensemble des positions p_s correspondant aux positions p_e , on applique le traitement général décrit plus haut (extraction des voisinages, composition des combinaisons d'impulsions, sélection de la combinaison optimale).

On retrouvera ce cas de durées de sous-trame égales mais de fréquences d'échantillonnage différentes sur les tables 5a à 5d ci-après, en référence à un exemple de réalisation où le codeur E est de type 3GPP NB-AMR et le codeur S est de type WB-AMR. Le codeur NB-AMR a une sous-trame de 40 échantillons pour une fréquence d'échantillonnage de 8 kHz. Le codeur WB-AMR utilise, quant à lui, 64 échantillons par sous-trame à 12,8 kHz. Dans les deux cas, la sous-trame a une durée de 5 ms. La table 5a donne la correspondance des positions dans une sous-trame du NB-AMR vers une sous-trame du WB-AMR, et la table 5b la correspondance inverse. Les tables de correspondance restreinte sont données dans les tables 5c et 5d.

NB-AMR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
WB-AMR	0	2	3	5	6	8	10	11	13	14	16	18	19	21	22	24	26	27	29	30	32	34	35	37	38	40	42	43	45	46	48	50	51	53	54	56	58	59	61	62

Table 5a: Table de correspondance temporelle du NB-AMR vers WB-AMR

WB-AMR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
NB-AMR	0	1	1	2	3	3	4	4	5	6	6	7	8	8	9	9	10	11	11	12	13	13	14	14	15	16	16	17	18	18	19	19
WB-AMR	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
NB-AMR	20	21	21	22	23	23	24	24	25	26	26	27	28	28	29	29	30	31	31	32	33	33	34	34	35	36	36	37	38	38	39	39

Table 5b: Table de correspondance temporelle du WB-AMR vers NB-AMR

Positions NB-AMR	0	1	2	3	4
Positions WB-AMR	0	2	3	5	6

Table 5c: Table de correspondance temporelle restreinte du NB-AMR vers WB-AMR

Positions WB-AMR	0	1	2	3	4	5	6	7
Positions NB-AMR	0	1	1	2	2	3	4	4

Table 5d: Table de correspondance temporelle restreinte du WB-AMR vers NB-AMR

En bref, en se référant à la figure 2a, on prévoit les étapes ci-après :

- a1) de quantification directe d'échelle temporelle de la première fréquence
 5 vers la seconde fréquence (étape 51 de la figure 2a),
 a2) et de détermination, en fonction de cette quantification, de chaque position
 d'impulsion dans une sous-trame au second format de codage caractérisé par
 la seconde fréquence d'échantillonnage, à partir d'une position d'impulsion
 dans une sous-trame au premier format de codage caractérisé par la première
 10 fréquence d'échantillonnage (étape 52 de la figure 2a).

En termes généraux, l'étape a1) de quantification s'effectue par calcul et/ou
 tabulation à partir d'une fonction qui, à une position p_e d'impulsion dans une
 sous-trame au premier format, fait correspondre une position p_s d'impulsion
 15 dans une sous-trame au second format, et cette fonction se présente
 sensiblement comme une combinaison linéaire faisant intervenir un coefficient
 multiplicatif correspondant au rapport de la seconde fréquence
 d'échantillonnage sur la première fréquence d'échantillonnage.

20 Par ailleurs, pour passer inversement d'une position d'impulsion dans une
 sous-trame au second format p_s à une position d'impulsion dans une sous-
 trame au premier format p_e , on applique bien entendu une fonction inverse de
 cette combinaison linéaire appliquée à une position d'impulsion dans une
 sous-trame au second format p_s .

25

On comprendra que le procédé de transcodage est complètement réversible et
 s'adapte aussi bien dans un sens de transcodage (E->S) que dans l'autre
 (S->E).

Dans un second mode de réalisation de l'adaptation des fréquences d'échantillonnage, on utilise un principe classique de changement de fréquence d'échantillonnage. On part de la sous-trame contenant les impulsions trouvées par le premier format. On sur-échantillonne à la fréquence
5 égale au plus petit commun multiple des deux fréquences d'échantillonnage F_e et F_s . Puis, après filtrage passe-bas, on sous-échantillonne pour revenir à la fréquence d'échantillonnage du second format, c'est-à-dire F_s . On obtient une sous-trame à la fréquence F_s contenant les impulsions de E filtrées. Là encore, on peut tabuler le résultat des opérations de sur-échantillonnage/filtrage
10 passe-bas/sous-échantillonnage pour chaque position possible d'une sous-trame de E. Ce traitement peut aussi s'effectuer par calcul "en ligne". Comme dans le premier mode de réalisation de l'adaptation des fréquences d'échantillonnage, on pourra associer une seule ou plusieurs positions de S à une position de E, comme expliqué ci-après, et appliquer le traitement général
15 au sens de l'invention décrit ci-avant.

Comme illustré dans la variante représentée sur la figure 2b, on prévoit les étapes ci-après :

- a'1) sur-échantillonner une sous-trame au premier format de codage
20 caractérisé par la première fréquence d'échantillonnage, à une fréquence F_{pcm} égale au plus petit commun multiple des première et seconde fréquences d'échantillonnage (étape 53 de la figure 2b), et
- a'2) appliquer à la sous-trame sur-échantillonnée un filtrage passe-bas (étape 54 de la figure 2b), suivi d'un sous-échantillonnage, pour atteindre une
25 fréquence d'échantillonnage correspondant à la seconde fréquence d'échantillonnage (étape 55 de la figure 2b).

Le procédé se poursuit par l'obtention, préférentiellement par seuillage, d'un nombre de positions éventuellement variable, ces positions étant adaptées des impulsions de E (étape 56) comme dans le premier mode de réalisation ci-
30 avant.

*** Fréquences d'échantillonnages égales mais durées de sous-trame différentes**

On décrit maintenant le traitement prévu dans le cas où les fréquences d'échantillonnage sont égales mais les durées de sous-trame sont différentes.

5 Cette situation correspond au cas "n" pour le test 23 mais "o" pour le test 22 de la figure 2. L'adaptation a) porte alors sur l'étape 33 de la figure 2.

Comme dans le cas ci-avant, on ne peut pas appliquer directement l'étape d'extraction des voisinages telle quelle. Il faut d'abord rendre compatibles les
10 deux sous-trames. Ici, les sous-trames sont différentes par leurs tailles. Face à cette incompatibilité, plutôt que de calculer les positions des impulsions comme le fait le tandem, une réalisation préférée propose une solution à faible complexité permettant de déterminer un répertoire restreint de combinaisons de positions pour les impulsions du second format à partir des positions des
15 impulsions du premier format. Cependant, la sous-trame de S et celle de E n'étant pas de même taille, il n'est pas possible d'établir une correspondance temporelle directe entre une sous-trame de S et une sous-trame de E. Comme le montre la figure 4 (dans laquelle les sous trames de E et S sont désignées respectivement par ST_E et ST_S), les frontières des sous-trames des deux
20 formats ne sont pas alignées et au cours du temps ces sous-trames se décalent l'une par rapport à l'autre.

Dans un mode de réalisation préféré, on propose de découper l'excitation de E en pseudo-sous-trames de la taille de celles de S et au rythme de S. Sur la
25 figure 5, on a représenté les pseudo sous-trames notées ST_E' . En pratique, cela revient aussi à effectuer une correspondance temporelle entre les positions dans les deux formats en prenant en compte la différence de taille des sous-trames pour aligner les positions relativement à une origine commune à E et S. La détermination de cette origine commune est décrite en
30 détail plus loin.

Une position p_e^o (respectivement p_s^o) du premier format (respectivement du second format) par rapport à cette origine coïncide avec la position p_e (respectivement p_s) de la sous-trame i_e (respectivement j_s) de E (respectivement S) relativement à cette sous-trame. On a ainsi :

$$5 \quad p_e^o = p_e + i_e L_e \text{ et } p_s^o = p_s + j_s L_s \text{ avec } 0 \leq p_e < L_e \text{ et } 0 \leq p_s < L_s$$

A une position p_e de la sous-trame i_e du format de E correspond la position p_s de la sous-trame j_s du format de S, p_s et j_s étant respectivement le reste et le quotient de la division euclidienne par L_s de la position p_e^o de p_e par rapport à une origine O commune à E et S :

$$j_s = \left\lfloor \frac{(p_e + i_e L_e)}{L_s} \right\rfloor \text{ et } p_s \equiv (p_e + i_e L_e) [L_s]$$

$$\text{avec } 0 \leq p_e < L_e \text{ et } 0 \leq p_s < L_s$$

$\lfloor \rfloor$ dénotant la partie entière, \equiv dénotant le modulo, l'indice d'une sous-trame de E (respectivement S) étant donné par rapport l'origine commune O.

Ainsi, les positions p_e situées dans une sous-trame j_s sont utilisées pour déterminer, selon le traitement général décrit ci-avant, un ensemble restreint de positions pour des impulsions de S dans la sous-trame j_s . Toutefois, lorsque $L_e > L_s$, il peut arriver qu'une sous-trame de S ne contienne aucune impulsion. Dans l'exemple de la figure 6, les impulsions de la sous-trame STE0 sont représentées par des traits verticaux. Le format de E peut très bien concentrer les impulsions de STE0 en fin de sous-trame si bien que la pseudo sous-trame STE'0 ne contient alors aucune impulsion. Toutes les impulsions placées par E se retrouvent dans STE'1 lors du découpage. Dans ce cas, une recherche focalisée classique est préférentiellement appliquée à la pseudo sous-trame STE'0.

On décrit maintenant des réalisations préférées pour la détermination d'une origine temporelle O commune aux deux formats. Cette référence commune

constitue la position (numéro 0) à partir de laquelle sont numérotées les positions des impulsions dans les sous-trames suivantes. Cette position 0 peut être définie de différentes manières, dépendant du système exploitant le procédé de transcodage au sens de la présente invention. Par exemple, pour
 5 un module de transcodage inclus dans un équipement d'un système de transmission, il sera naturel de prendre pour origine la première position de la première trame reçue après la mise en route de l'équipement.

Toutefois, l'inconvénient de ce choix est que les positions prennent des
 10 valeurs de plus en grandes et il peut devenir nécessaire de les limiter. Pour cela, il suffit de remettre à jour la position de l'origine commune à chaque fois que cela est possible. Ainsi, si les longueurs respectives L_e et L_s des sous-trames de E et S, sont constantes au cours du temps, la position de l'origine commune est remise à jour à chaque fois que les frontières des sous-trames
 15 de E et S sont alignées. Cela arrive périodiquement, la période (en échantillons) étant égale au plus petit commun multiple de L_e et L_s .

On peut aussi envisager le cas où L_e et/ou L_s ne sont pas constantes dans le temps. Il n'est plus possible de trouver un multiple commun aux deux
 20 longueurs de sous-trame, notées à présent $L_e(n)$ et $L_s(n)$, n représentant le numéro de la sous-trame. Dans ce cas, il convient de sommer au fur et à mesure les valeurs $L_e(n)$ et $L_s(n)$ et comparer à chaque sous-trame les deux sommes obtenues :

$$25 \quad T_e(k) = \sum_{n=1}^k L_e(n) \quad \text{et} \quad T_s(k') = \sum_{n=1}^{k'} L_s(n)$$

A chaque fois que l'on a $T_e(k) = T_s(k')$, l'origine commune est remise à jour (et prise à la position $k \times L_e$ ou encore à $k' \times L_s$). Quant aux deux sommes T_e et T_s , elles sont préférentiellement réinitialisées.

En bref et plus généralement, en appelant première (respectivement seconde) durée de sous-trame la durée de sous-trame du premier (respectivement second) format de codage, les étapes d'adaptation effectuées lorsque les durées de sous-trame sont différentes sont résumées sur la figure 7 et sont préférentiellement les suivantes :

a20) définition d'une origine commune O aux sous-frames des premier et second formats (étape 70),

a21) découpage des sous-frames successives du premier format de codage caractérisé par une première durée de sous-trame, pour former des pseudo-sous-frames de durées L'_e correspondantes à la seconde durée de sous-trame (étape 71),

a22) mise à jour de l'origine commune O (étape 79),

a23) et détermination de correspondance entre les positions d'impulsions dans les pseudo-sous-frames p'_e et dans les sous-frames au second format (étape 80).

Préférentiellement, pour déterminer l'origine commune O, on discrimine au test 72 de la figure 7 les cas suivants :

- les première et seconde durées sont fixes dans le temps (sortie "o" du test 72), et
- les première et seconde durées varient dans le temps (sortie "n" du test 72).

Dans le premier cas, la position temporelle de l'origine commune est remise périodiquement à jour (étape 74) à chaque instant où des frontières de sous-frames respectives de première durée $St(L_e)$ et de seconde durée $St(L_s)$ sont alignées dans le temps (test 73 effectué sur ces frontières).

Dans le second cas, préférentiellement :

a221) on effectue successivement les deux sommations respectives des sous-frames au premier format $T_e(k)$ et des sous-frames au second format $T_s(k')$ (étape 76),

5 a222) on détecte une occurrence d'une égalité entre lesdites deux sommes, définissant un instant de remise à jour de ladite origine commune (test 77),

a223) on réinitialise les deux sommes précitées (étape 78), après ladite occurrence, pour une future détection d'une prochaine origine commune.

10 Maintenant, dans le cas où les durées de sous-trame et les fréquences d'échantillonnage sont différentes, il suffit de combiner judicieusement les algorithmes de correspondance entre les positions de E et S décrites dans les deux cas précédents.

15 * EXEMPLES DE REALISATION

On décrit maintenant trois exemples de réalisation du transcodage au sens de l'invention. Ces exemples de réalisation décrivent la mise en œuvre des traitements prévus dans les cas présentés ci-avant dans des codeurs de parole à analyse par synthèse normalisés. Les deux premiers modes illustrent
20 le cas favorable où les fréquences d'échantillonnage, comme les durées des sous-frames, sont identiques. Le dernier exemple illustre le cas où les durées des sous-frames sont différentes.

* Exemple de réalisation n°1

25 Le premier exemple de réalisation est appliqué au transcodage intelligent entre le modèle MP-MLQ du G.723.1 à 6,3 kbit/s et le modèle ACELP à 4 impulsions du G.723.1 à 5,3 kbit/s.

30 Un transcodage intelligent du débit haut vers le débit bas du G.723.1 met en présence un modèle MP-MLQ à 6 et 5 impulsions avec un modèle ACELP à

4 impulsions. L'exemple de réalisation présenté ici permet de déterminer les positions des 4 impulsions de l'ACELP à partir des positions des impulsions du MP-MLQ.

5 On rappelle ci-après le fonctionnement du codeur G.723.1.

Le codeur multi-débits UIT-T G.723.1 et ses répertoires multi-impulsionnels ont été présentés ci-avant. Il est précisé seulement qu'une trame du G.723.1 comporte 240 échantillons à 8 kHz, et qu'elle est divisée en quatre sous-
10 trames de 60 échantillons. La même restriction est imposée aux positions des impulsions de tout vecteur-code de chacun des trois dictionnaires multi-impulsionnels. Ces positions doivent toutes avoir la même parité (toutes paires ou toutes impaires). La sous-trame de 60(+4) positions est ainsi découpée en deux grilles de 32 positions. La grille paire comporte les positions numérotées
15 [0, 2, 4,..., 58, (60,62)]. La grille impaire comporte les positions [1, 3, 5,..., 59,(61,63)]. Pour chaque débit, l'exploration du répertoire, même non exhaustive, reste complexe comme indiqué précédemment.

On décrit maintenant la sélection d'un sous-ensemble du répertoire ACELP du
20 G.723.1 à 5,3 kbit/s à partir d'un élément d'un répertoire MP-MLQ du G.723.1 à 6,3 kbit/s.

On cherche à modéliser le signal d'innovation d'une sous-trame par un élément du répertoire ACELP du G.723.1 à 5,3 kbit/s connaissant l'élément du
25 répertoire MP-MLQ du G.723.1 à 6,3 kbit/s déterminé lors d'un premier codage. On dispose donc des N_e positions ($N_e=5$ ou 6) des impulsions choisies par le codeur G.723.1 à 6,3 kbit/s.

Par exemple, on peut supposer que les positions extraites du train binaire du codeur G.723.1 à 6,3kbit/s pour une sous-trame dont l'excitation est modélisée par $N_e=5$ impulsions sont :

$$e_0 = 0; \quad e_1 = 8; \quad e_2 = 28; \quad e_3 = 38; \quad e_4 = 46;$$

5 On rappelle qu'aucune adaptation des fréquences d'échantillonnage ou des durées de sous-frames n'est à faire ici. Après cette étape de récupération des positions e_i , une étape suivante consiste alors à extraire directement les voisinages droits et gauches de ces 5 impulsions. Les voisinages droits et gauches sont pris ici égaux à 2. L'ensemble P_s des positions sélectionnées est :

$$P_s = \{-2, -1, 0, 1, 2\} \cup \{6, 7, 8, 9, 10\} \cup \{26, 27, 28, 29, 30\} \cup \{36, 37, 38, 39, 40\} \cup \{44, 45, 46, 47, 48\}$$

La troisième étape consiste à composer l'ensemble restreint des positions possibles pour chaque impulsion (ici une piste) du répertoire ACELP du G.723.1 à 5,3 kbit/s en prenant $N_s=4$ intersections de P_s avec les 4 ensembles des positions des pistes paires (respectivement impaires) autorisées par ce dernier répertoire (comme représenté dans la table 1).

Pour la parité paire :

$$S_0 = P_s \cap \{0, 8, 16, \dots, 56\}; \quad S_1 = P_s \cap \{2, 10, 18, \dots, 58\}; \quad S_2 = P_s \cap \{4, 12, 20, \dots, 52, (60)\}; \\ S_3 = P_s \cap \{6, 14, 22, \dots, 54, (62)\};$$

20 d'où : $S_0 = \{0, 8, 40, 48\}; \quad S_1 = \{2, 10, 26\}; \quad S_2 = \{28, 36, 44\}; \quad S_3 = \{6, 30, 38, 46\};$

Pour la parité impaire :

$$S_0 = P_s \cap \{1, 9, \dots, 57\}; \quad S_1 = P_s \cap \{3, 11, \dots, 59\}; \quad S_2 = P_s \cap \{5, 13, \dots, 53, (61)\}; \\ S_3 = P_s \cap \{7, 15, \dots, 55, (63)\};$$

d'où : $S_0 = \{1, 9\}; \quad S_1 = \{27\}; \quad S_2 = \{29, 37, 45\}; \quad S_3 = \{7, 39, 47\};$

25

La combinaison de ces positions sélectionnées constitue le nouveau répertoire restreint dans lequel s'effectuera la recherche. Pour cette dernière étape, la procédure de sélection du jeu de positions optimales s'appuie sur le critère

CELP comme le fait le G.723.1 au mode 5,3 kbit/s. L'exploration peut être exhaustive ou, de préférence, focalisée.

Le nombre de combinaisons de positions dans le répertoire restreint est égal à 180 ($=4*3*3*4+2*1*3*3$) au lieu des 8192 ($=2*8*8*8*8$) combinaisons de positions du répertoire ACELP du G.723.1 à 5,3 kbit/s.

On indique qu'on peut restreindre davantage le nombre de combinaisons en ne considérant que la parité choisie au mode 6,3 kbit/s (dans l'exemple cité la parité paire). Dans ce cas, le nombre de combinaisons du répertoire restreint est égal à 144.

Il peut arriver (selon les taille des voisinages considérées) que pour une des quatre impulsions, l'ensemble P_s ne contienne aucune position pour une piste du modèle ACELP (cas où l'un des ensemble S_i est vide). Ainsi, pour des voisinages de taille 2, quand les positions des N_e impulsions sont toutes sur la même piste, P_s ne contient que des positions de cette piste et des pistes adjacentes. Dans ce cas, selon le compromis qualité/complexité désiré, il est possible soit de remplacer l'ensemble S_i par T_i (ce qui revient à ne pas restreindre l'ensemble des positions de cette piste), soit d'augmenter le voisinage droit (ou gauche) des impulsions. Par exemple, si toutes les impulsions du code à 6,3 kbit/s sont sur la piste 2, avec des voisinages droit et gauche égaux à 2, la piste 0 n'aura pas de positions quelle que soit la parité. Il suffit alors d'augmenter de 2 la taille du voisinage gauche et/ou droit pour attribuer des positions à cette piste 0.

Pour illustrer cette réalisation, on part de l'exemple suivant :

$$e_0 = 4; \quad e_1 = 12; \quad e_2 = 20; \quad e_3 = 36; \quad e_4 = 52;$$

L'ensemble P_s des positions sélectionnées est :

$$P_s = \{2,3,4,5,6\} \cup \{10,11,12,13,14\} \cup \{18,19,20,21,22\} \cup \{34,35,36,37,38\} \cup \{50,51,52,53,54\}$$

En supposant que l'on veuille garder la même parité, la répartition initiale de ces positions pour les 4 impulsions est :

$S_0 = \emptyset$; $S_1 = \{2, 10, 18, 34, 50\}$; $S_2 = \{4, 12, 20, 36, 52\}$; $S_3 = \{6, 14, 22, 38, 54\}$;

En augmentant de 2 le voisinage gauche des impulsions, on obtient :

5 $S_0 = \{0, 8, 16, 32, 48\}$; $S_1 = \{2, 10, 18, 34, 50\}$;

$S_2 = \{4, 12, 20, 36, 52\}$; $S_3 = \{6, 14, 22, 38, 54\}$

(avec donc $S_0 \neq \emptyset$).

* Exemple de réalisation n°2

10 Le second exemple suivant illustre l'application de l'invention aux transcodages intelligents entre des modèles ACELP de même longueur. En particulier, ce second exemple de réalisation est appliqué au transcodage intelligent entre l'ACELP à 4 impulsions du G.729 à 8 kbit/s et l'ACELP à 2 impulsions du G.729 à 6,4 bits/s.

15 Un transcodage intelligent entre les modes 6,4 kbit/s et 8 kbit/s du codeur G.729, met en présence un répertoire ACELP à deux impulsions et un second à quatre impulsions. L'exemple présenté ici permet de déterminer les positions de 4 impulsions (8kbit/s) à partir des positions de 2 impulsions (6,4 kbit/s) et
20 réciproquement.

On rappelle succinctement le fonctionnement du codeur UIT-T G.729. Ce codeur peut fonctionner à trois débits: 6,4; 8 et 11,8 kbit/s. On considère donc ici les deux premiers débits. Une trame de G.729 comporte 80 échantillons à
25 8 kHz. Cette trame est divisée en deux sous-trames de 40 échantillons. Pour chaque sous-trame, le G.729 modélise le signal d'innovation par des impulsions selon le modèle ACELP. Il en utilise quatre au mode 8 kbit/s et deux pour le mode 6,4 kbit/s. Les tables 2 et 4 ci-avant donnent les positions que peuvent prendre les impulsions pour ces deux débits. A 6,4 kbit/s, une

recherche exhaustive de toutes les combinaisons (512) des positions est effectuée. A 8 kbit/s, une recherche focalisée est préférentiellement utilisée.

Le traitement général au sens de l'invention est encore utilisé ici. Toutefois et de façon avantageuse, on tire partie ici de la structure ACELP commune aux deux répertoires. La mise en correspondance des jeux de positions exploite ainsi une découpe de la sous-trame de 40 échantillons en 5 pistes de 8 positions, donnés dans la table 6 ci-après.

Pistes	Positions
P ₀	0, 5, 10, 15, 20, 25, 30, 35
P ₁	1, 6, 11, 16, 21, 26, 31, 36
P ₂	2, 7, 12, 17, 22, 27, 32, 37
P ₃	3, 8, 13, 18, 23, 28, 33, 38
P ₄	4, 9, 14, 19, 24, 29, 34, 39

Table 6: Répartition des positions en cinq pistes dans les dictionnaires ACELP du G.729

Dans les deux répertoires, les positions des impulsions se partagent ces pistes, comme le montre la table 7 ci-après.

Toutes les impulsions sont caractérisées par leur piste et leur rang dans cette piste. Le mode à 8 kbit/s place une impulsion sur chacune des trois premières pistes et la dernière impulsion sur l'une des deux dernières pistes. Le mode à 6,4 kbit/s quant à lui place sa première impulsion sur les pistes P₁ ou P₃, et sa deuxième impulsion sur les pistes P₀, P₁, P₂ ou P₄.

Mode	Impulsions	Pistes
6,4 kbit/s	i_0	P_1, P_3
	i_1	P_0, P_1, P_2, P_4
8 kbit/s	i_0	P_0
	i_1	P_1
	i_2	P_2
	i_3	P_3, P_4

Table 7: Répartition des impulsions des répertoires ACELP du G.729 à 8 et 6,4 kbit/s dans les cinq pistes.

L'entrelacement des pistes (structure ISSP) est exploité dans cette réalisation pour faciliter l'extraction des voisinages et la composition des sous-ensembles restreints de positions. Ainsi, pour se déplacer d'une piste à une autre, il suffit de se décaler d'une unité vers la droite ou vers la gauche. Par exemple, si l'on se place à la 5^{ème} position de la piste 2 (de position absolue 22), un décalage de 1 vers la droite (+1) fait passer à la 5^{ème} position de la piste 3 (de position absolue 23) et un décalage vers la gauche (-1) fait passer à la 5^{ème} position de la piste 1 (de position absolue 21).

Plus généralement, un décalage de $\pm d$ d'une position se traduit ici par les effets suivants.

15 Au niveau des pistes P_i :

voisinage droit : $P_i \Rightarrow P_{(i+d) \equiv 5}$

voisinage gauche : $P_i \Rightarrow P_{(i-d) \equiv 5}$

Au niveau du rang m dans la piste :

* voisinage droit :

20 si $(i+d) \leq 4$: $m_i \Rightarrow m_i$

sinon: $m_i \Rightarrow m_i + 1$

* voisinage gauche :

si $(i-d) \geq 0$: $m_i \Rightarrow m_i$

sinon: $m_i \Rightarrow m_i - 1$

On décrit maintenant la sélection d'un sous-ensemble du répertoire ACELP à 4
5 impulsions du codeur G.729 à 8 kbit/s à partir d'un élément d'un répertoire
ACELP à 2 impulsions du codeur G.729 à 6,4 kbit/s.

On considère une sous-trame de G.729 au mode 6,4 kbit/s. Deux impulsions
sont placées par ce codeur mais il faut déterminer les positions des autres
10 impulsions que doit placer le G.729 à 8 kbit/s. Pour restreindre radicalement la
complexité, une seule position par impulsion est choisie et une seule
combinaison de positions est retenue. Avantageusement, l'étape de sélection
est donc immédiate. On choisit deux des quatre impulsions du G.729 à 8 kbit/s
aux mêmes positions que celles du mode à 6,4 kbit/s, puis on place les deux
15 impulsions restantes à proximité immédiate des deux premières. Comme
indiqué ci-avant, la structure en pistes est exploitée. A la première étape de
récupération des deux positions par décodage de l'indice binaire (sur 9 bits)
des 2 positions, les deux pistes correspondantes sont aussi déterminées. A
partir de ces deux pistes (éventuellement identiques), les trois dernières
20 étapes d'extraction des voisinages, de composition des sous-ensembles
restreints et de sélection d'une combinaison d'impulsions sont alors
judicieusement associées. On distingue plusieurs cas selon les pistes P_i ($i=0$ à
4) sur lesquelles se trouvent les deux impulsions du mode à 6,4 kbit/s.

25 On note e_k les positions des impulsions du mode à 6,4 kbit/s et s_k celle du
mode à 8 kbit/s. La table 8 ci-après présente pour chacun des cas les
positions choisies. Les colonnes notées " $P_{j+d}=P_i$ " précisent la loi de voisinage
au niveau des pistes et aboutissant à la piste P_i . On rappelle qu'au niveau des
pistes P_i :

30 * pour le voisinage droit : $P_i \Rightarrow P_{(i+d) \equiv 5}$

* pour le voisinage gauche : $P_i \Rightarrow P_{(i-d) \equiv 5}$

e_0 (Piste)	e_1 (Piste)		s_0		s_1		s_2		s_3	
			Position	$P_{i+d}=P_0$	Position	$P_{i+d}=P_1$	Position	$P_{i+d}=P_2$	Position	$P_{i+d}=P_3/P_4$
P_1	$e_0=e_1$	P_1	e_1-1	P_1-1	E_1	P_1	e_1+1	P_1+1	e_1+2	P_1+2
	$e_0 \neq e_1$		e_0-1	P_1-1	E_0	P_1	e_1+1	P_1+1	e_1+2	P_1+2
P_1		P_0	e_1	P_0	E_0	P_1	e_0+1	P_1+1	$e_1-1^{(1)}$	$P_0^{(1)}-1$
P_1		P_2	e_0-1	P_1-1	E_0	P_1	e_1	P_2	e_1+1	P_2+1
P_1		P_4	$e_1+1^{(2)}$	$P_4^{(2)}+1$	E_0	P_1	e_0+1	P_1+1	e_1	P_4
P_3		P_0	e_1	P_0	E_1+1	P_0+1	e_0-1	P_3-1	e_0	P_3
P_3		P_1	e_1-1	P_1-1	E_1	P_1	e_0-1	P_3-1	e_0	P_3
P_3		P_2	$e_0+2^{(3)}$	$P_3^{(3)}+2$	E_1-1	P_2-1	e_1	P_2	e_0	P_3
P_3		P_4	$e_1+1^{(4)}$	$P_4^{(4)}+1$	E_0-2	P_3-2	e_0-1	P_3-1	e_1	P_4

Table 8: Choix du répertoire restreint du G.729 à 8 kit/s à partir des deux impulsions du répertoire ACELP du G.729 à 6,4 kbit/s.

Préférentiellement, on cherche donc à équilibrer la répartition des 4 positions par rapport aux deux positions de départ, mais on indique qu'un autre choix peut être effectué. Quatre cas (signalés par un exposant entre parenthèses dans la table 8) peuvent poser toutefois des problèmes d'effets de bord:

Cas (1) : si $e_1 = 0$, alors on ne peut pas prendre $s_3=e_1-1$. On choisira $s_3= e_0+2$.

Cas (2) : si $e_1 = 39$, alors on ne peut pas prendre $s_0=e_1+1$. On choisit $s_0= e_0 -1$.

Cas (3): si $e_1 = 38$, alors on ne peut pas prendre $s_0=e_0+2$. On choisit $s_0= e_1 -2$.

Cas (4): si $e_1 = 39$, alors on ne peut pas prendre $s_0=e_1+1$. On choisit $s_0= e_0 -3$.

Pour diminuer encore la complexité, le signe de chaque impulsion s_k peut être pris égal à celui de l'impulsion e_j de laquelle elle est déduite.

On décrit maintenant la sélection d'un sous-ensemble du répertoire ACELP à 2 impulsions du G.729 à 6,4 kbit/s à partir d'un élément d'un répertoire ACELP à 4 impulsions du G.729 à 8 kbit/s.

Pour une sous-trame de G.729 au mode à 8 kbit/s, la première étape est la récupération des positions des quatre impulsions générées par le mode 8 kbit/s. Le décodage de l'indice binaire (sur 13 bits) des 4 positions permet d'obtenir leur rang dans leur piste respective pour les trois premières positions
 5 (des pistes 0 à 2) et la piste (3 ou 4) de la quatrième impulsion ainsi que son rang dans cette piste. Chaque position e_i ($0 \leq i < 4$) est caractérisée par le couple (p_i, m_i) où p_i est l'indice de sa piste et m_i son rang dans cette piste. On a:

$$e_i = 5m_i + p_i, \text{ avec } 0 \leq m_i < 8 \text{ et } p_i = i \text{ pour } i < 3 \text{ et } p_3 = 3 \text{ ou } 4.$$

10

Comme déjà mentionné, l'extraction des voisinages et la composition des sous-ensembles restreints sont combinées et exploitent avantageusement la structure ISSP commune aux deux répertoires. En exploitant la propriété de positions voisines induite par l'entrelacement des pistes, on construit les cinq
 15 intersections T_j de l'ensemble P_s des voisinages des 4 positions avec les 5 pistes P_j .

$$T_j = P_s \cap P_j$$

Ainsi, un voisin droit (respectivement gauche) de +1 (respectivement -1) de l'impulsion (p, m) appartient à T_{p+1} si $p < 4$ (respectivement à T_{p-1} si $p > 0$), sinon
 20 (cas $p=4$) à T_0 à condition que $m < 7$ (respectivement à T_4 (cas $i=0$) à condition que $m > 0$). La restriction sur le voisin droit pour une position de la quatrième impulsion appartenant à la quatrième piste (respectivement gauche pour une position de la première piste) permet de s'assurer que la position voisine n'est pas en dehors de la sous-trame.

25

Ainsi, en utilisant la notation modulo 5 ($\equiv 5$), un voisin droit (respectivement gauche) de +1 (respectivement -1) de l'impulsion (p, m) appartient à $T'_{(p+1) \equiv 5}$ (respectivement à $T'_{(p-1) \equiv 5}$). On rappelle qu'il faut tenir compte des effets de bord. En généralisant à une taille d de voisinage, un voisin droit de $+d$
 30 (respectivement gauche de $-d$) de l'impulsion (p, m) appartient à $T'_{(p+d) \equiv 5}$ (respectivement à $T'_{(p-d) \equiv 5}$). Le rang du voisin à $\pm d$ est égal à m si $p+d \leq 4$ (ou $p-$

$d \geq 0$), sinon le rang m est incrémenté pour un voisin droit et décrémenté pour un voisin gauche. La prise en compte des effets de bord revient donc à s'assurer que: $m < 7$ si $p+d > 4$ et que $m > 0$ si $p-d < 0$.

A partir de cette répartition des voisins dans les 5 pistes, il est simple de
 5 déterminer les sous-ensembles restreints S_0 et S_1 des positions des deux impulsions :

$$S_0 = T'_1 \cup T'_3 \text{ et } S_1 = T'_0 \cup T'_1 \cup T'_2 \cup T'_4$$

La quatrième et dernière étape consiste à effectuer la recherche du couple optimal dans les deux sous-ensembles obtenus. L'algorithme de recherche
 10 (comme celui normalisé exploitant la structure en pistes) et le rangement par piste des impulsions simplifient là encore l'algorithme de recherche. En pratique, il est donc inutile de constituer explicitement les sous-ensembles restreints S_0 et S_1 car les ensembles T'_j peuvent être utilisés seuls.

15 Dans l'exemple ci-après, le mode à 8 kbit/s du G.729 a placé ses quatre impulsions aux positions suivantes :

$$e_0 = 5 ; e_1 = 21 ; e_2 = 22 ; e_3 = 34 ;$$

Ces 4 positions sont caractérisées par les 4 couples $(p_i, m_i) = (0, 1), (1, 4), (2, 4), (4, 6)$.

20 Si l'on prend un voisinage fixe égal à 1, on construit les 5 intersections T'_j comme suit :

$e_0 : (0, 1)$ donne : $(4, 0)$ à gauche et $(1, 1)$ à droite

$e_1 : (1, 4)$ donne : $(0, 4)$ à gauche et $(2, 4)$ à droite

$e_2 : (2, 4)$ donne : $(1, 4)$ à gauche et $(3, 4)$ à droite

25 $e_3 : (4, 6)$ donne : $(3, 6)$ à gauche et $(0, 7)$ à droite

On a donc :

$$T'_0 = \{(0, 1), (0, 4), (0, 7)\}; T'_1 = \{(1, 4), (1, 1)\}; T'_2 = \{(2, 4)\}; T'_3 = \{(3, 4), (3, 6)\}; \\ T'_4 = \{(4, 6), (4, 0)\}$$

En revenant à la notation par positions :

$$T'_0=\{5,20,35\}; T'_1=\{21, 6\}; T'_2=\{22\}; T'_3=\{23,33\};$$

$$T'_4 = \{34,4\}$$

5 A la dernière étape, un algorithme semblable à celui du G.729 à 6,4 kbit/s effectue la recherche du meilleur couple d'impulsions. Cet algorithme est ici bien moins complexe car le nombre de combinaisons de positions à explorer est très restreint. Dans l'exemple, il n'y a que 4 (= Cardinal(T'_1) + Cardinal(T'_3))
10 fois 8 (= Cardinal(T'_0) + Cardinal(T'_1) + Cardinal(T'_2) + Cardinal(T'_4)) combinaisons à tester, soit 32 combinaisons au lieu de 512.

Pour un voisinage de taille 1, moins de 8% des combinaisons des positions est à explorer en moyenne sans dépasser 10% (50 combinaisons). Pour un
15 voisinage de taille 2, moins de 17% des combinaisons des positions est à explorer en moyenne et au plus 25% des combinaisons est à explorer. Pour un voisinage de taille 2, la complexité du traitement proposé dans l'invention (en cumulant le coût de la recherche dans le répertoire restreint au coût de l'extraction des voisinages associé à la composition des intersections)
20 représente moins de 30% d'une recherche exhaustive pour une qualité équivalente.

*** Exemple de réalisation n°3**

Le dernier exemple illustre les passages entre le modèle ACELP du G.729 à 8 kbit/s et le modèle MP-MLQ du G.723.1 à 6,3 kbit/s.

25 Un transcodage intelligent des impulsions entre le G.723.1 (mode 6,3 kbit/s) et le G.729 (mode 8 kbit/s) comporte deux difficultés importantes. En premier lieu, la taille des trames est différente (40 échantillons pour le G.729 contre 60 échantillons pour le G.723.1). La seconde difficulté est liée à la structure
30 différente des dictionnaires, du type ACELP pour le G.729 et du type MP-MLQ pour le G.723.1. L'exemple de réalisation présenté ici montre comment

l'invention lève ces deux difficultés afin de transcoder à moindre coût les impulsions tout en préservant la qualité du transcodage.

On effectue d'abord une correspondance temporelle entre les positions dans les deux formats en prenant en compte la différence de taille des sous-frames pour aligner les positions relativement à une origine commune à E et S. Les longueurs des sous-frames du G.729 et du G.723.1 ayant comme plus petit commun multiple 120, la correspondance temporelle est effectuée par blocs de 120 échantillons soit deux sous-frames de G.723.1 pour trois sous-frames de G.729, comme le montre l'exemple de la figure 4b. En variante, on peut préférer travailler sur des blocs de trames complètes. Dans ce cas, on choisit des blocs de 240 échantillons soit une trame de G.723.1 (4 sous-frames) pour trois trames de G.729 (6 sous-frames).

On décrit maintenant la sélection d'un sous-ensemble du répertoire MP-MLQ du G.723.1 à 6,3 kbit/s à partir d'éléments du répertoire ACELP à 4 impulsions du G.729 à 8 kbit/s. La première étape consiste à récupérer les positions des impulsions par blocs de 3 sous-frames (d'index i_e , $0 \leq i_e \leq 2$) du G.729. On note $p_e(i_e)$ une position de la sous-trame i_e de ce bloc.

Avant d'extraire les voisinages, on convertit ces 12 positions $p_e(i_e)$ en 12 positions notées $p_s(j_s)$, réparties dans deux sous-frames (d'index j_s , $0 \leq j_s \leq 1$) de G.723.1. On peut utiliser la relation générale ci-avant (faisant intervenir le modulo de la longueur de sous-trame) pour réaliser l'adaptation des durées de sous-frames. On préfère toutefois ici distinguer simplement trois cas selon l'indice i_e :

si $i_e = 0$, alors $j_s = 0$ et $p_s = p_e$
 si $i_e = 2$, alors $j_s = 1$ et $p_s = p_e + 20$
 si $i_e = 1$, alors si $p_e < 20$ $j_s = 0$ et $p_s = p_e + 40$, sinon ($p_e \geq 20$) : $j_s = 1$ et $p_s = p_e - 20$

On n'effectue ainsi ni division ni opération modulo n .

Les 4 positions récupérées dans la sous-trame STE0 du bloc sont directement attribuées à la sous-trame STS0 avec la même position, celles de la sous-trame STE2 du bloc sont directement attribuées à la sous-trame STS1 avec un incrément de +20 de position, les positions de la sous-trame STE1 inférieurs à 20 sont attribuées à la sous-trame STS0 avec un incrément de +40, les autres étant attribuées à la sous-trame STS1 avec un décrétement de -20.

On extrait ensuite les voisinages de ces 12 positions. Il faut noter que l'on peut autoriser les voisinages droits (respectivement gauches) des positions de la sous-trame STS0 (respectivement STS1) à sortir de leur sous-trame, ces positions voisines étant alors dans la sous-trame STS1 (respectivement STS0).

L'étape de correspondance temporelle et d'extraction de voisinage peuvent être inversées. Dans ce cas, on peut autoriser les voisinages droits (respectivement gauches) des positions de la sous-trame STE0 (respectivement STE2) à sortir de leur sous-trame, ces positions voisines étant alors dans la sous-trame STE1. De même, les voisinages droits (respectivement gauches) des positions dans STE1, peuvent conduire à des positions voisines dans STE2 (respectivement STE0).

Une fois l'ensemble des positions restreintes pour chaque sous-trame STS constitué, la dernière étape consiste à explorer pour chaque sous-trame STS son répertoire restreint ainsi constitué pour sélectionner les N_p (6 ou 5) impulsions de même parité. Cette procédure peut être dérivée de l'algorithme normalisé ou s'inspirer d'autres procédures de focalisation.

Pour illustrer cet exemple de réalisation, on considère trois sous-frames du G.729 qui vont permettre de construire les sous-répertoires de deux sous-frames du G.723.1. En supposant que le G.729 donne les positions suivantes :

$$STE0 : e_{00} = 5 ; e_{01} = 1 ; e_{02} = 32 ; e_{03} = 39 ;$$

$$STE1 : e_{10} = 15 ; e_{11} = 31 ; e_{12} = 22 ; e_{13} = 4 ;$$

$$STE2 : e_{20} = 0 ; e_{21} = 1 ; e_{22} = 37 ; e_{23} = 24 ,$$

après application de l'étape ci-avant de correspondance temporelle, l'attribution de ces 12 positions aux sous-trames STS0 et STS1 est :

$$STS0 : s_{00} = 5 ; s_{01} = 1 ; s_{02} = 32 ; s_{03} = 39 ; (s_{0k} = e_{0k})$$

$$STS0 : s'_{10} = 55 ; s'_{13} = 44 ; (s'_{0k} = e_{1k} + 40, \text{ si } e_{1k} < 20)$$

$$STS1 : s'_{11} = 11 ; s'_{12} = 2 (s'_{1k} = e_{1k} - 20, \text{ si } e_{1k} \geq 20)$$

$$STS1 : s_{20} = 20 ; s_{21} = 21 ; s_{22} = 57 ; s_{23} = 44 ; (s_{0k} = e_{2k} + 20)$$

On a donc les jeux de positions {1, 5, 32, 39, 44, 55} pour la sous-trame STS0 et {2, 11, 20, 21, 44, 57} pour la sous-trame STS1.

Il faut à présent en extraire les voisinages. En prenant par exemple un voisinage fixé à 1, on obtient :

$$P_{s0} = \{0,1,2\} \cup \{4,5,6\} \cup \{31,32,33\} \cup \{38,39,40\} \cup \{43,44,45\} \cup \{54,55,56\}$$

$$P_{s1} = \{1,2,3\} \cup \{10,11,12\} \cup \{20,21,22\} \cup \{21,22,23\} \cup \{43,44,45\} \cup \{56,57,58\}$$

Le MP-MLQ n'impose aucune contrainte aux impulsions, mis à part leur parité. Sur une sous-trame, elles doivent être toutes de la même parité. Il faut donc ici scinder P_{s0} et P_{s1} en deux sous ensembles, avec :

$$- P_{s0} : \{0,2,4,6,32,40,44,54,56\} \text{ et } \{1,5,31,33,39,43,45,55\}$$

$$- P_{s1} : \{2,10,12,20,22,44,56\} \text{ et } \{1,3,11,21,23,43,45,57\}$$

Ce sous-répertoire est enfin transmis à l'algorithme de sélection qui détermine les N_p meilleures positions au sens du critère CELP pour les sous-trames STS0 et STS1 du G.723.1. On restreint ainsi considérablement le nombre de combinaisons testées. En effet, par exemple dans la sous-trame STS0, il reste 9 positions paires et 8 positions impaires au lieu de 30 et 30.

On s'assure toutefois de certaines précautions dans les cas où les positions choisies par le G.729 sont telles que l'extraction des voisinages donne un

nombre N de positions possibles inférieur au nombre de positions du G.723.1 ($N < N_p$). C'est le cas notamment lorsque les positions du G.729 se suivent toutes (ex : {0,1,2,3}). On prévoit alors deux possibilités :

- 5 - augmenter la taille du voisinage pour les sous-trames concernées jusqu'à obtenir une taille suffisante pour P_s (taille $\geq N_p$),
- ou sélectionner les N premières impulsions et autoriser pour les $N_p - N$ impulsions restantes une recherche parmi les $30 - N$ positions restantes de la grille, comme décrit ci-avant.

10 On décrit maintenant le traitement inverse consistant à sélectionner un sous-ensemble du répertoire ACELP à 4 impulsions du G.729 à 8 kbit/s à partir d'éléments d'un répertoire MP-MLQ du G.723.1 à 6,3 kbit/s.

15 Globalement, le traitement est similaire. Deux sous-trames de G.723.1 correspondent à 3 trames de G.729. Ici aussi, on extrait les positions du G.723.1 que l'on traduit dans l'échelle de temps du G.729. On pourra avantageusement traduire ces positions sous la forme "*piste - rang dans la piste*" afin de profiter comme précédemment de la structure ACELP pour extraire les voisinages et rechercher les positions optimales.

20

Les mêmes aménagements que précédemment sont prévus pour prévenir les cas où l'extraction des voisinages donnerait des positions en nombre insuffisant (ici moins de 4 positions).

25 Ainsi, la présente invention permet de déterminer à moindre coût les positions d'un jeu d'impulsions à partir d'un premier jeu d'impulsions, les deux jeux d'impulsions appartenant à deux répertoires multi-impulsionnels. Ces deux répertoires peuvent se distinguer par leur taille, la longueur et le nombre d'impulsions de leurs mots de codes ainsi que par les règles régissant les
30 positions et/ou amplitudes des impulsions. On privilégie les voisinages des positions des impulsions du (ou des) jeu(x) choisi(s) dans le premier répertoire

pour déterminer celles d'un jeu dans le deuxième répertoire. L'invention permet en plus d'exploiter la structure des répertoires de départ et/ou d'arrivée pour réduire davantage la complexité. A travers le premier exemple ci-avant présentant le passage d'un modèle MP-MLQ à un modèle ACELP, on

5 comprendra que l'invention s'applique aisément à deux modèles multi-impulsionnels ayant des contraintes structurelles différentes. A travers le second exemple de réalisation présentant le passage entre deux modèles possédant un nombre d'impulsions différent mais basés sur la même structure ACELP, on comprendra que l'invention permet avantageusement l'exploitation

10 de la structure des répertoires pour réduire la complexité de transcodage. A travers le troisième exemple présentant le passage entre un modèle MP-MLQ et un modèle ACELP, on comprendra que l'invention peut s'appliquer même pour des codeurs de longueurs de sous-frames ou de fréquences d'échantillonnage différentes. L'invention permet d'ajuster le compromis

15 qualité/complexité et, en particulier, de réduire fortement la complexité de calcul pour une dégradation minime par rapport à une recherche classique d'un modèle multi-impulsionnel.

Revendications

1. Procédé de transcodage entre un premier codeur/décodeur en compression et au moins un second codeur/décodeur en compression, lesdits premier et
5 second codeurs/décodeurs étant de type impulsionnel et utilisant des dictionnaires multi-impulsionnels dans lesquels chaque impulsion comporte une position repérée par un indice associé,

caractérisé en ce qu'il comporte les étapes suivantes :

a) le cas échéant, adaptation de paramètres de codage entre lesdits premier
10 et second codeurs/décodeurs,

b) obtention, à partir du premier codeur/décodeur, d'un nombre choisi (N_e) de positions d'impulsions et d'indices de positions (e_i) respectivement associés,

c) pour chaque position d'impulsion courante d'indice donné, formation d'un groupe de positions d'impulsions comportant au moins la position d'impulsion
15 courante et des positions d'impulsions d'indices associés immédiatement inférieurs et immédiatement supérieurs à l'indice donné,

d) sélection, en fonction de positions d'impulsions (T_j) admises par le second codeur/décodeur, d'une partie au moins des positions d'impulsions dans un ensemble (P_s) constitué par une union desdits groupes formés à l'étape c), et

20 e) transmission des positions d'impulsions ainsi sélectionnées au second codeur/décodeur, pour un codage/décodage à partir desdites positions transmises,

ladite sélection de l'étape d) faisant alors intervenir un nombre de positions d'impulsions possibles inférieur au nombre total de positions d'impulsions
25 possibles du dictionnaire du second codeur/décodeur.

2. Procédé selon la revendication 1, dans lequel le premier codeur/décodeur (E) utilise, dans un premier format de codage, un premier nombre d'impulsions,

30 caractérisé en ce que ledit nombre (N_e) choisi à l'étape b) correspond à ce premier nombre de positions d'impulsions.

3. Procédé selon la revendication 2, dans lequel :

- le premier codeur/décodeur (E) utilise, dans un premier format de codage, un premier nombre (N_e) de positions d'impulsions,
- 5 - le second codeur/décodeur (S) utilise, dans un second format de codage, un second nombre (N_s) de positions d'impulsions, caractérisé en ce qu'il comporte en outre une étape consistant à discriminer au moins les cas où :
- le premier nombre (N_e) est supérieur ou égal au second nombre (N_s),
- 10 - le premier nombre (N_e) est inférieur au second nombre (N_s).

4. Procédé selon la revendication 3, dans lequel le premier nombre (N_e) est supérieur ou égal au second nombre (N_s) ($N_e \geq N_s$), caractérisé en ce que chaque groupe formé à l'étape c) comporte des positions d'impulsions voisines de droite (v_d^j) et des positions d'impulsions voisines de gauche (v_g^j) de ladite position d'impulsion courante d'indice donné, et en ce que les nombres respectifs de positions d'impulsions voisines de gauche et de droite sont choisis en fonction d'un compromis en complexité et en qualité de transcodage.

20

5. Procédé selon la revendication 4, caractérisé en ce qu'à l'étape d), on construit un sous-répertoire de combinaisons de positions d'impulsions résultant des intersections (S_j) de :

- l'ensemble (P_s) constitué par une union desdits groupes formés à l'étape c),
- 25 et
- des positions d'impulsions (T_j) admises par le second codeur/décodeur, de sorte que ledit sous-répertoire soit de taille inférieure au nombre de combinaisons de positions d'impulsions (T_j) admises par le second codeur/décodeur.

30

6. Procédé selon la revendication 5, caractérisé en ce que, après l'étape e), on recherche dans ledit sous-répertoire un jeu de positions optimal comportant ledit second nombre (N_s) de positions, au niveau du second codeur (S).

5 7. Procédé selon la revendication 6, caractérisé en ce que l'étape de recherche du jeu de positions optimal est mise en œuvre par une recherche focalisée pour accélérer l'exploration dudit sous-répertoire.

10 8. Procédé selon l'une des revendications précédentes, dans lequel ledit premier codeur/décodeur est agencé pour délivrer une succession de trames codées, caractérisé en ce que les nombres respectifs de positions d'impulsions dans les groupes formés à l'étape c) sont successivement choisis d'une trame à l'autre.

15 9. Procédé selon la revendication 3, dans lequel le premier nombre (N_e) est inférieur au second nombre (N_s) ($N_e < N_s$),
caractérisé en ce que l'on teste en outre si les positions d'impulsions prévues dans le second nombre de positions d'impulsions (N_s) se retrouvent parmi des positions d'impulsions des groupes formés à l'étape c),
20 et, en cas de réponse négative à ce test, on augmente le nombre de positions d'impulsions dans les groupes formés à l'étape c).

10. Procédé selon la revendication 3, caractérisé en ce que l'on discrimine en outre le cas où le second nombre N_s est compris entre une et deux fois le
25 premier nombre N_e ($N_e < N_s < 2N_e$),
et, dans ce cas :
c1) on sélectionne d'emblée les N_e positions d'impulsions,
c2) et on sélectionne en outre un nombre complémentaire de positions d'impulsions $N_s - N_e$, définies dans le voisinage immédiat des positions
30 d'impulsions sélectionnées à l'étape c1).

11. Procédé selon l'une des revendications précédentes, dans lequel ledit premier codeur/décodeur est agencé pour opérer avec une première fréquence d'échantillonnage donnée et à partir d'une première durée donnée de sous-trame,

5 caractérisé en ce que lesdits paramètres de codage pour lesquels est réalisée ladite adaptation de l'étape a) comportent au moins une durée de sous-trame et une fréquence d'échantillonnage, tandis que le second codeur/décodeur opère avec une seconde fréquence d'échantillonnage et une seconde durée de sous-trame,

10 et en ce que, à l'étape a), on discrimine les quatre cas suivants :

- les première et seconde durées sont égales et les première et seconde fréquences sont égales,
- les première et seconde durées sont égales et les première et seconde fréquences sont différentes,
- 15 - les première et seconde durées sont différentes et les première et seconde fréquences sont égales,
- les première et seconde durées sont différentes et les première et seconde fréquences sont différentes.

20 12. Procédé selon la revendication 11, dans lequel les première et seconde durées sont égales et les première et seconde fréquences d'échantillonnage sont différentes,

caractérisé en ce qu'il comporte des étapes :

a1) de quantification directe d'échelle temporelle de la première fréquence
25 vers la seconde fréquence,

a2) et de détermination, en fonction de ladite quantification, de chaque position d'impulsion dans une sous-trame au second format de codage caractérisé par la seconde fréquence d'échantillonnage, à partir d'une position d'impulsion dans une sous-trame au premier format de codage caractérisé par la première
30 fréquence d'échantillonnage.

13. Procédé selon la revendication 12, caractérisé en ce que l'étape a1) de quantification s'effectue par calcul et/ou tabulation à partir d'une fonction qui, à une position d'impulsion dans une sous-trame au premier format (p_e), fait correspondre une position d'impulsion dans une sous-trame au second format (p_s), ladite fonction se présentant sensiblement comme une combinaison linéaire faisant intervenir un coefficient multiplicatif correspondant au rapport de la seconde fréquence d'échantillonnage sur la première fréquence d'échantillonnage.
14. Procédé selon la revendication 13, caractérisé en ce que, pour passer inversement d'une position d'impulsion dans une sous-trame au second format (p_s) à une position d'impulsion dans une sous-trame au premier format (p_e), on applique une fonction inverse de la ladite combinaison linéaire appliquée à une position d'impulsion dans une sous-trame au second format (p_s).
15. Procédé selon la revendication 11, dans lequel les première et seconde durées sont égales et les première et seconde fréquences d'échantillonnage sont différentes,
- caractérisé en ce qu'il comporte les étapes suivantes :
- a'1) sur-échantillonner une sous-trame au premier format de codage caractérisé par la première fréquence d'échantillonnage, à une fréquence égale au plus petit commun multiple des première et seconde fréquences d'échantillonnage, et
- a'2) appliquer à la sous-trame sur-échantillonnée un filtrage passe-bas, suivi d'un sous-échantillonnage, pour atteindre une fréquence d'échantillonnage correspondant à la seconde fréquence d'échantillonnage.
16. Procédé selon la revendication 15, caractérisé en ce que le procédé se poursuit par l'obtention, par seuillage, d'un nombre de positions, le cas échéant, variable.

17. Procédé selon la revendication 12, caractérisé en ce qu'il comporte en outre une étape de mise en correspondance, pour chaque position (p_e) d'une impulsion d'une sous-trame au premier format de codage caractérisé par la première fréquence d'échantillonnage, d'un groupe de positions d'impulsions (p_s) dans une sous-trame au second format de codage caractérisé par la seconde fréquence d'échantillonnage, chaque groupe comportant un nombre de positions qui est fonction du rapport entre la seconde et la première fréquence d'échantillonnage (F_s/F_e).
18. Procédé selon la revendication 11, dans lequel les première et seconde durées de sous-frames sont différentes,
caractérisé en ce qu'il comporte les étapes :
- a20) de définition d'une origine commune (O) aux sous-frames des premier et second formats,
 - a21) de découpage des sous-frames successives du premier format de codage caractérisé par une première durée de sous-trame, pour former des pseudo-sous-frames de durées correspondantes à la durée de sous-trame du second format,
 - a22) de mise à jour de ladite origine commune,
 - a23) et d'une détermination de correspondance entre les positions d'impulsions dans les pseudo-sous-frames et dans les sous-frames au second format.
19. Procédé selon la revendication 18, caractérisé en ce que l'on discrimine en outre les cas suivants :
- les première et seconde durées sont fixes dans le temps, et
 - les première et seconde durées varient dans le temps.
20. Procédé selon la revendication 19, dans lequel les première et seconde durées sont fixes dans le temps,

caractérisé en ce que la position temporelle de ladite origine commune est remise périodiquement à jour à chaque instant où des frontières de sous-frames respectives de première et de seconde durée sont alignées dans le temps.

5

21. Procédé selon la revendication 19, dans lequel les première et seconde durées varient dans le temps,

caractérisé en ce que :

10 a221) on effectue successivement deux sommations respectives des durées des sous-frames au premier format et des durées des sous-frames au second format,

a222) on détecte une occurrence d'une égalité entre lesdites deux sommes, définissant un instant de remise à jour de ladite origine commune,

15 a223) on réinitialise lesdites deux sommes, après ladite occurrence, pour une future détection d'une prochaine origine commune.

22. Produit programme d'ordinateur destiné à être stocké dans une mémoire d'une unité de traitement, notamment d'un ordinateur ou d'un terminal mobile, ou sur un support mémoire amovible et destiné à coopérer avec un lecteur de
20 l'unité de traitement,

caractérisé en ce qu'il comporte des instructions pour la mise en œuvre du procédé de transcodage selon l'une des revendications précédentes.

23. Dispositif de transcodage entre un premier codeur/décodeur en
25 compression et au moins un second codeur/décodeur en compression, lesdits premier et second codeurs/décodeurs étant de type impulsionnel et utilisant des dictionnaires multi-impulsionnels dans lesquels chaque impulsion comporte une position repérée par un indice associé,
caractérisé en ce qu'il comporte une mémoire propre à stocker des instructions
30 d'un produit programme d'ordinateur selon la revendication 22.

FIG.1a.

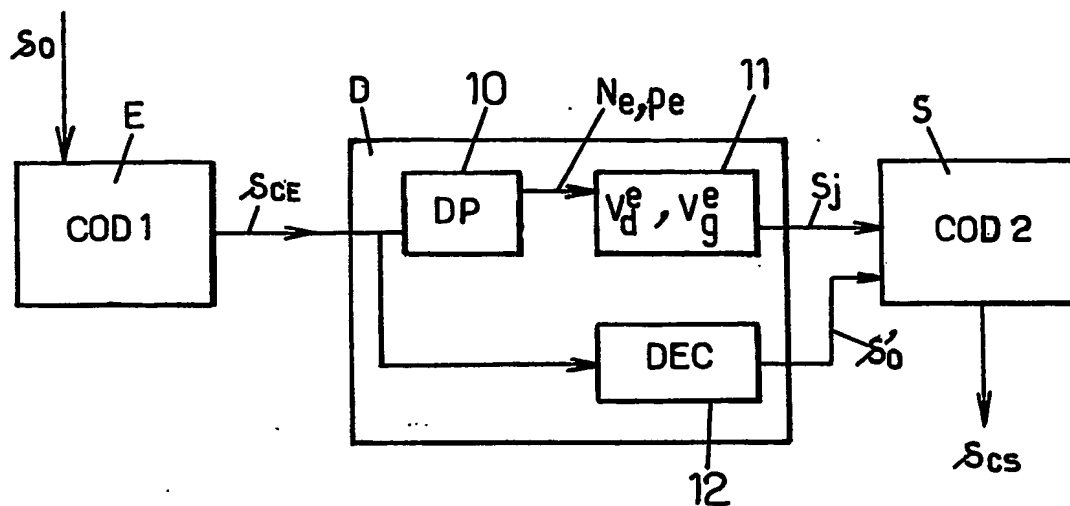


FIG.1b.

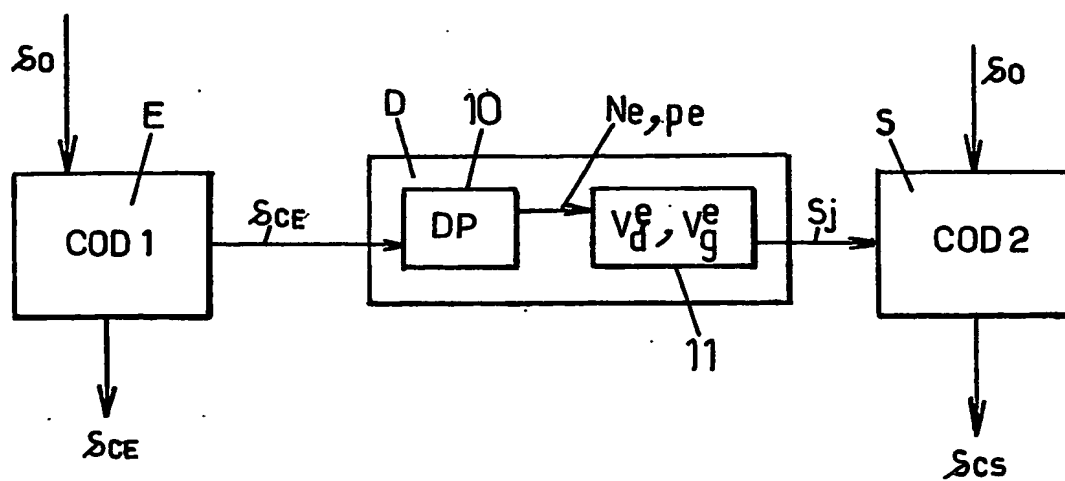
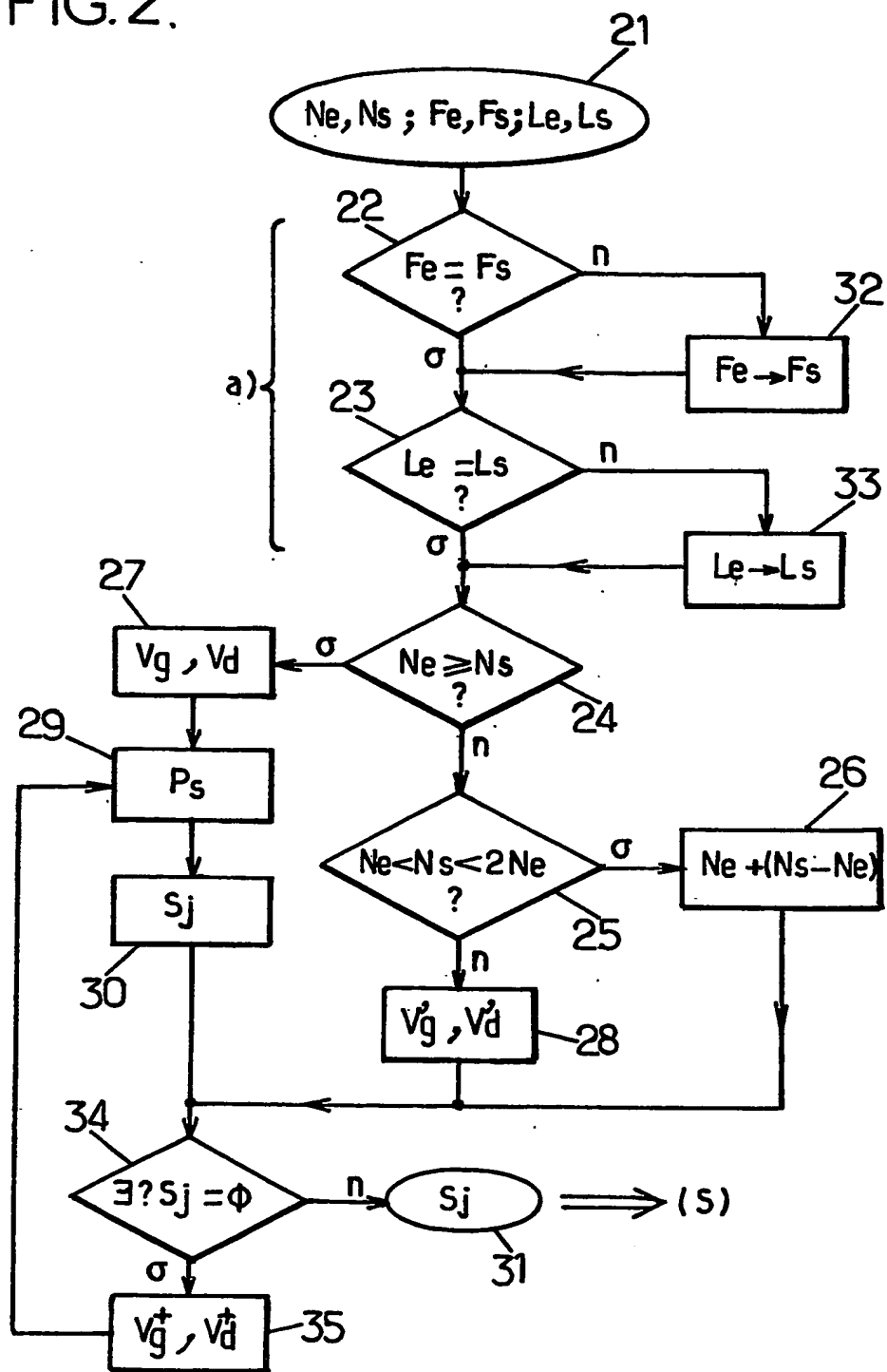


FIG. 2.



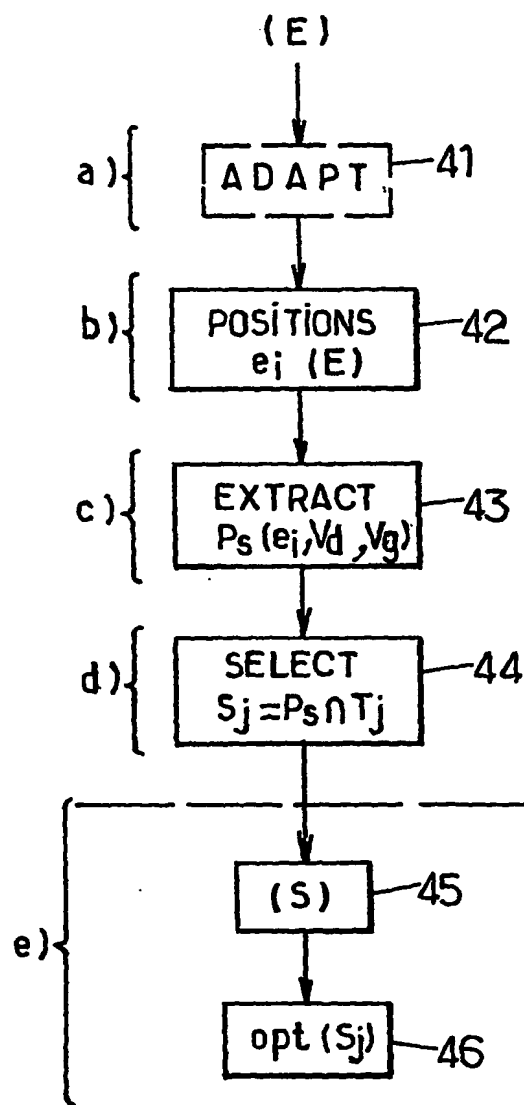
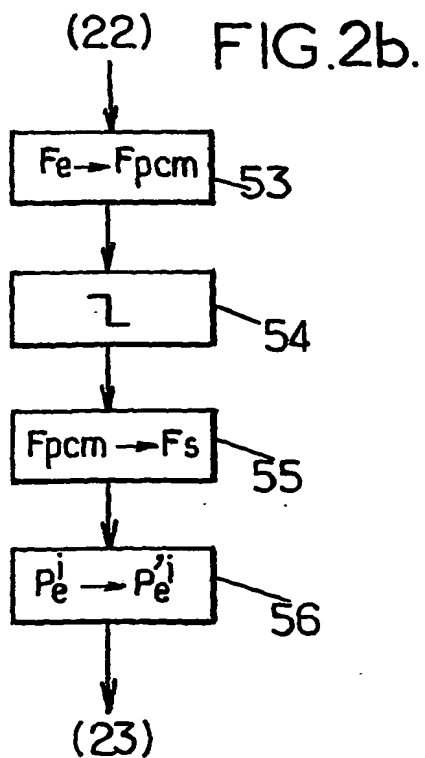
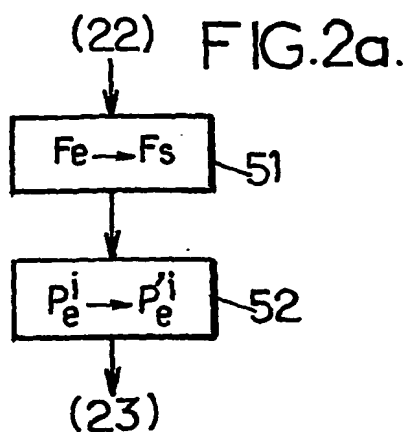


FIG.3.

FIG.4.

(E)

STe0		STe1		STe2
STs0	STs1	STs2	STs3	STs4

(S)

FIG.5.

(E)

STe0		STe1		STe2
STe0	STe1	STe1		STe2
STs0	STs1	STs2	STs3	STs4

(S)

FIG.6.

(E)

STe0		STe1
STe0	STe1	STe1

(S)

FIG.4b.

G.723.1

ST0		ST1
ST0	ST1	ST3

G.729

5/5

FIG.7.

